



*Open Tools from Sybase, Inc.*

**InfoMaker**

***User's Guide***

*Version 6*

**InfoMaker<sup>®</sup>**

AA0537

October 1997

Copyright © 1991-1997 Sybase, Inc. and its subsidiaries.

All rights reserved.

Printed in Ireland.

Information in this manual may change without notice and does not represent a commitment on the part of Sybase, Inc. and its subsidiaries.

The software described in this manual is provided by Powersoft Corporation under a Powersoft License agreement. The software may be used only in accordance with the terms of the agreement.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc. and its subsidiaries.

Sybase, Inc. and its subsidiaries claim copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of other rights of Sybase, Inc. and its subsidiaries.

ClearConnect, Column Design, ComponentPack, InfoMaker, ObjectCycle, PowerBuilder, PowerDesigner, Powersoft, S-Designor, SQL SMART, and Sybase are registered trademarks of Sybase, Inc. and its subsidiaries. Adaptive Component Architecture, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Warehouse, AppModeler, DataArchitect, DataExpress, Data Pipeline, DataWindow, dbQueue, ImpactNow, InstaHelp, Jaguar CTS, jConnect for JDBC, MetaWorks, NetImpact, Optima++, Power++, PowerAMC, PowerBuilder Foundation Class Library, Power J, PowerScript, PowerSite, Powersoft Portfolio, Powersoft Professional, PowerTips, ProcessAnalyst, Runtime Kit for Unicode, SQL Anywhere, The Model For Client/Server Solutions, The Future Is Wide Open, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, Viewer, WarehouseArchitect, Watcom, Watcom SQL Server, Web.PB, and Web.SQL are trademarks of Sybase, Inc. or its subsidiaries. Certified PowerBuilder Developer and CPD are service marks of Sybase, Inc. or its subsidiaries. DataWindow is a patented proprietary technology of Sybase, Inc. or its subsidiaries.

AccuFonts is a trademark of AccuWare Business Solutions Ltd.

All other trademarks are the property of their respective owners.

# Contents

About This Book .....	xix
-----------------------	-----

## PART 1

### THE INFOMAKER ENVIRONMENT

<b>1</b>	<b>Working with InfoMaker.....</b>	<b>3</b>
	What is InfoMaker? .....	4
	Customizing InfoMaker at installation.....	5
	What you can do using InfoMaker.....	7
	Reports.....	7
	Queries.....	9
	Forms .....	10
	Data pipelines.....	12
	Applications .....	12
	Database management .....	13
	The InfoMaker environment .....	14
	About the PowerBar .....	14
	About PowerTips .....	15
	About the PowerPanel.....	15
	Painter summary .....	16
	Opening a painter or tool.....	16
	Using online Help .....	18
	Linking to the online books.....	18
	Using popup menus .....	20
	Using property sheets .....	22
	Using toolbars .....	24
	Toolbar basics .....	24
	Dropdown toolbars .....	24
	Controlling the display of toolbars .....	25
	Moving toolbars using the mouse.....	25
	Customizing toolbars.....	26
	Creating new toolbars .....	31
	Deleting toolbars .....	32
	Using the file editor .....	33

	Setting file editing properties .....	33
	Editing activities.....	34
	Executing AppleScript scripts on the Macintosh .....	35
	Changing fonts .....	37
	Defining colors .....	38
	Managing the initialization file .....	39
	About the initialization file.....	39
	Starting InfoMaker from the command line .....	42
<b>2</b>	<b>Managing Your Environment.....</b>	<b>45</b>
	About managing your environment .....	46
	About InfoMaker libraries and special files.....	47
	Working in the Environment painter workspace .....	48
	Displaying dates and times .....	49
	Displaying comments .....	49
	Refreshing the library list.....	49
	Modifying comments .....	50
	Accessing objects .....	52
	Opening an object .....	52
	Previewing an object .....	52
	Setting the current library .....	53
	Creating a new library .....	54
	Copying, moving, and deleting objects in a library .....	55
	Copying and moving objects in a library.....	55
	Deleting objects in a library .....	56
	Accessing shared queries stored on a network .....	58
	Defining libraries for user-defined form styles .....	59
	Query Governor .....	61
	Accessing the Query Governor .....	62
	Using the Query Governor .....	62
<b>3</b>	<b>Managing the Database.....</b>	<b>65</b>
	About databases .....	66
	About tables and columns .....	66
	About keys.....	66
	About indexes.....	67
	About views .....	67
	About extended attributes .....	68
	About managing databases .....	69
	How you work with the database.....	69
	Using the Database painter.....	72
	Changing the database connection.....	72
	About the painter .....	72

Modifying database preferences .....	77
Creating and deleting a SQL Anywhere database .....	79
Working with tables in the Database painter .....	81
Opening a table .....	81
Modifying properties in the Database painter .....	83
Closing a table .....	86
Dropping a table .....	86
Working with tables in the Table painter .....	88
About the Table painter .....	88
Creating a new table from scratch in the Table painter .....	88
Creating a new table from an existing table in the Table painter .....	90
Specifying column definitions and extended attributes .....	91
Modifying table properties in the Table painter .....	94
Altering a table in the Table painter .....	95
Cutting, copying, and pasting columns in the Table painter .....	97
Specifying fonts for the table in the Table painter .....	98
Working in SQL Syntax view .....	99
Logging applied SQL syntax changes .....	100
Printing the table definition .....	101
Working with keys .....	102
Why you should use keys .....	102
What you can do in the painters .....	102
Viewing keys .....	103
Opening related tables .....	104
Defining primary keys .....	104
Defining foreign keys .....	105
Dropping a key .....	107
Working with indexes .....	109
Creating an index .....	109
Modifying an index .....	110
Dropping an index .....	110
Working with views .....	112
Opening a view .....	112
Creating a view .....	113
Displaying a view's SQL statement .....	114
Joining tables .....	115
Dropping a view .....	117
Manipulating data .....	118
Opening the Data Manipulation painter .....	118
Retrieving data .....	119
Modifying data .....	119
Sorting and filtering data .....	120
Viewing row information .....	123

Importing data .....	123
Printing data .....	124
Saving data .....	124
Returning to the Database painter workspace .....	125
Administering the database.....	126
About administering databases .....	126
Opening the Database Administration painter.....	127
Using the editor .....	127
Building and executing SQL statements .....	128

## **PART 2**

## **REPORTS**

<b>4</b>	<b>Defining Reports.....</b>	<b>135</b>
	Introducing reports .....	136
	Report examples .....	136
	Reports versus DataWindow objects .....	142
	Building a report .....	143
	Connecting to a database .....	143
	Modifying an existing report .....	143
	Creating a new report.....	143
	Choosing a presentation style .....	145
	Using the Tabular style.....	146
	Using the Freeform style .....	146
	Using the Grid style .....	147
	Using the Label style .....	147
	Using the N-Up style .....	149
	Using the Group presentation style .....	151
	Using the Composite presentation style.....	151
	Using the Graph and Crosstab presentation styles.....	151
	Using the OLE 2.0 presentation style.....	151
	Using the RichText presentation style.....	152
	Choosing report-wide options .....	153
	Defining the data source .....	155
	How to choose the data source.....	155
	Using Quick Select.....	156
	Using SQL Select.....	166
	Using Query .....	180
	Using External .....	181
	Using Stored Procedure.....	182
	Generating and saving a report.....	185
	About the Powersoft repository and reports .....	185
	Saving the report.....	186
	Defining queries .....	188

	Previewing the query .....	188
	Saving the query .....	189
	Naming the query .....	189
	Modifying a query .....	190
	What's next.....	191
<b>5</b>	<b>Enhancing Reports .....</b>	<b>193</b>
	Working in the Report painter workspace .....	195
	Understanding the Report painter workspace .....	195
	Using the Report painter toolbars.....	198
	Using property sheets in the Report painter.....	199
	Selecting objects in the Report painter.....	200
	Using keyboard shortcuts in the Report painter .....	202
	Resizing bands in the Report painter workspace .....	204
	Using zoom in the Report painter .....	204
	Undoing changes in the Report painter .....	204
	Previewing a report .....	205
	Retrieving data .....	206
	Sorting and filtering data .....	209
	Importing data into a report .....	209
	Using preview.....	210
	Printing data .....	211
	Saving data in an external file .....	212
	Saving the data in HTML Table format.....	213
	Working with PSR files .....	215
	Mailing reports.....	217
	Working in a grid report .....	218
	Modifying general report properties .....	220
	Changing the report style .....	220
	Setting colors in a report .....	221
	Specifying properties of a grid report .....	222
	Specifying pointers for a report .....	222
	Defining print specifications for a report .....	223
	Modifying text in a report .....	227
	Naming objects in a report .....	228
	Using borders in a report.....	229
	Specifying variable-height detail bands in a report.....	229
	Modifying the data source of a report.....	232
	Reorganizing objects in a report .....	235
	Displaying boundaries for objects in a report .....	235
	Using the grid and the ruler in a report.....	236
	Deleting objects in a report.....	236
	Moving objects in a report .....	237
	Copying objects in a report.....	237

Resizing objects in a report .....	238
Aligning objects in a report .....	238
Equalizing the space between objects in a report .....	239
Equalizing the size of objects in a report .....	239
Sliding objects to remove blank space in a report .....	240
Prompting for retrieval criteria in a report .....	242
Adding objects to a report .....	244
Adding columns to a report .....	244
Adding text to a report .....	245
Adding drawing objects to a report .....	245
Adding a groupbox to a report .....	246
Adding pictures to a report .....	246
Adding computed fields to a report .....	248
Adding buttons to a report .....	253
Adding graphs to a report .....	255
Adding OLE objects to a report .....	255
Adding reports to a report .....	255
Positioning objects in a report .....	256
Storing data in a report .....	257
What happens when you preview the report .....	258
Retrieving rows as needed .....	259
Saving retrieved rows to disk .....	260

<b>6</b>	<b>Displaying and Validating Data .....</b>	<b>261</b>
	About displaying and validating data .....	262
	Presenting the data .....	262
	Validating data .....	263
	Working with display formats .....	264
	Using display formats .....	265
	Defining display formats .....	269
	Working with edit styles .....	276
	Using edit styles .....	277
	Defining edit styles .....	280
	Defining a code table .....	288
	Working with validation rules .....	292
	Understanding validation rules .....	292
	Using validation rules .....	293
	Defining validation rule properties in the Database painter ...	293
	Defining a validation rule in the Form painter .....	297
	Maintaining the entities .....	300

<b>7</b>	<b>Filtering, Sorting, and Grouping Rows .....</b>	<b>301</b>
	Filtering rows .....	302



	Sorting rows .....	305
	Suppressing repeating values .....	306
	Grouping rows .....	308
	Using the Group presentation style .....	310
	Defining groups in an existing report .....	314
<b>8</b>	<b>Highlighting Information in Reports and Forms .....</b>	<b>325</b>
	Overview of highlighting information .....	326
	Conditionally modifying properties at execution time .....	330
	Example 1: creating a gray bar effect .....	332
	Example 2: rotating objects .....	335
	Example 3: highlighting rows of data .....	337
	Example 4: changing the size and location of objects .....	340
	Supplying property values .....	343
	Background.Color .....	344
	Border .....	345
	Brush.Color .....	346
	Brush.Hatch .....	346
	Color .....	349
	Font.Escapement (for rotating objects) .....	349
	Font.Height .....	352
	Font.Italic .....	353
	Font.Strikethrough .....	354
	Font.Underline .....	355
	Font.Weight .....	355
	Format .....	356
	Height .....	357
	Pen.Color .....	357
	Pen.Style .....	358
	Pen.Width .....	359
	Pointer .....	360
	Visible .....	360
	Width .....	361
	X .....	362
	X1, X2 .....	362
	Y .....	363
	Y1, Y2 .....	363
	Specifying colors .....	364
<b>9</b>	<b>Using Nested Reports .....</b>	<b>367</b>
	About nested reports .....	368
	Creating a report using the Composite presentation style .....	372
	Placing a nested report in another report .....	374

Placing a related nested report in another report .....	374
Placing an unrelated nested report in another report .....	378
Working with nested reports .....	379
Adjusting nested report width .....	380
Using the Autosize Height option for nested reports .....	380
Changing a nested report from one report to another .....	381
Modifying the contents of a nested report .....	382
Adding another nested report to a composite report .....	382
Supplying retrieval arguments to relate a nested report to its base report .....	383
Specifying criteria to relate a nested report to its base report .....	385
Using the Slide option for a nested report .....	387
Using the Start On New Page option (composite only) .....	387
Using the Trail the Footer option (composite only) .....	387

<b>10</b>	<b>Working with Graphs .....</b>	<b>389</b>
	Overview of graphs .....	390
	Parts of a graph .....	390
	Types of graphs .....	392
	Using graphs in reports .....	396
	Placing a graph in a report .....	396
	Using the graph's property sheet .....	398
	Changing a graph's position and size .....	399
	Associating data with a graph .....	401
	Using overlays .....	412
	Using the Graph presentation style .....	414
	Defining a graph's properties .....	415
	Using the Graph property page of the graph's property sheet .....	415
	Naming a graph .....	416
	Defining a graph's title .....	416
	Specifying the type of graph .....	417
	Using legends .....	417
	Sorting data .....	417
	Specifying text properties for titles, labels, axes, and legends .....	418
	Specifying overlap and spacing .....	422
	Specifying axis properties .....	423
	Specifying a border .....	427
	Specifying a pointer .....	428
	Specifying point of view in 3D graphs .....	428

<b>11</b>	<b>Working with Crosstabs</b> .....	<b>431</b>
	About crosstabs .....	432
	Two types of crosstabs.....	434
	Creating crosstabs .....	436
	Associating data with a crosstab .....	437
	Specifying the information .....	437
	What happens .....	439
	Specifying more than one row or column .....	441
	Previewing crosstabs .....	443
	Enhancing crosstabs .....	444
	Specifying basic properties .....	444
	Modifying the data associated with the crosstab .....	445
	Changing the names used for the columns and rows .....	446
	Defining summary statistics.....	447
	Cross-tabulating ranges of values .....	450
	Creating static crosstabs .....	453
	Using property conditional expressions.....	454
 <b>12</b>	 <b>Working with Rich Text</b> .....	 <b>457</b>
	What is rich text?.....	458
	Using the RichText presentation style.....	459
	Creating the report .....	460
	Formatting for RichText objects within the report.....	463
	Previewing and printing .....	469
	Formatting keys and toolbars.....	471
 <b>13</b>	 <b>Using OLE in a Report</b> .....	 <b>475</b>
	OLE support in reports .....	476
	About activation.....	476
	OLE objects and OLE presentation style .....	478
	Adding an OLE object to the report .....	479
	Using the OLE presentation style .....	479
	Defining the OLE object .....	480
	Previewing the report .....	482
	Specifying data for the OLE object.....	484
	Using OLE columns in a report .....	489
	Creating an OLE column .....	489
	Embedding reports in other applications.....	494
	Creating a new InfoMaker report from within another application .....	494
	Adding a report saved as a PSR file .....	495

**PART 3**

**DATA PIPELINES**

<b>14</b>	<b>Working with Data Pipelines.....</b>	<b>499</b>
	About data pipelines.....	500
	Defining a data pipeline.....	501
	Piping extended attributes.....	502
	Creating a data pipeline.....	503
	Modifying the data pipeline definition.....	506
	Items you can modify in the workspace.....	506
	Choosing a pipeline operation.....	508
	Dependency of modifications on pipeline operation.....	509
	When execution stops.....	511
	Piping blob data.....	513
	Changing the destination and source databases.....	514
	Correcting pipeline errors.....	516
	Saving a pipeline.....	518
	Using an existing pipeline.....	519
	Pipeline examples.....	520

**PART 4**

**FORMS**

<b>15</b>	<b>Defining Forms .....</b>	<b>525</b>
	About forms.....	526
	About the Form painter's PainterBar.....	528
	About creating new forms.....	528
	Freeform forms.....	530
	Grid forms.....	531
	Master/Detail One-To-Many forms.....	532
	Master/Detail Many-To-One forms.....	534
	Creating and saving forms.....	536
	Creating basic forms.....	536
	Creating a master/detail form.....	537
	Defining data so a form can update a database.....	542
	Generating and saving forms.....	543
	Working with forms.....	546
	Running forms.....	546
	Limiting the retrieved data.....	547
	Importing data into a form.....	548
	Saving data in an external file.....	549
	Printing forms.....	550
	Actions in forms.....	550
	Accessing and deleting existing forms.....	553

	Controlling updates .....	555
	What you can do .....	555
	Specifying the table to update .....	556
	Specifying the unique key columns .....	557
	Specifying an identity column .....	557
	Specifying updatable columns .....	558
	Specifying the WHERE clause for update/delete .....	558
	Specifying update when key is modified .....	561
<b>16</b>	<b>Enhancing Forms.....</b>	<b>563</b>
	About enhancing forms .....	564
	Working in the Form painter workspace.....	565
	Using the Form painter toolbars .....	565
	Using the popup menus in the Form painter .....	567
	Using property sheets in the Form painter .....	568
	Using keyboard shortcuts in the Form painter workspace ....	568
	Selecting objects in the Form painter .....	570
	Defining default colors and borders in the Form painter .....	571
	Printing the form definition.....	573
	Reorganizing objects in the form.....	574
	Using the grid in the Form painter .....	574
	Deleting objects in the Form painter.....	575
	Moving objects in the Form painter .....	576
	Copying and pasting objects in the Form painter .....	576
	Resizing objects in the Form painter .....	577
	Aligning objects in the Form painter .....	578
	Equalizing the space between objects in the Form painter.....	579
	Equalizing the size of objects in the Form painter .....	579
	Undoing changes in the Form painter .....	580
	Sliding objects in a form .....	580
	Modifying general form properties.....	582
	Specifying a title for a form .....	582
	Setting colors for a form .....	583
	Specifying the display of scrollbars for a form .....	584
	Specifying pointers for a form.....	585
	Modifying text in a form .....	586
	Defining the tab order in a form .....	587
	Using borders in a form .....	589
	Prompting for retrieval criteria in a form .....	589
	Modifying the data source of a form .....	592
	Adding objects to the form .....	594
	Adding columns to a form.....	594
	Adding text to a form .....	594

Adding computed fields to a form .....	595
Adding pictures to a form .....	599
Adding command buttons to a form .....	600
Adding picture buttons to a form .....	603
Adding reports to a form .....	605
Adding drawing objects to a form .....	606
Highlighting information in a form .....	608
Displaying and validating data in a form .....	610

**PART 5**

**APPLICATIONS**

<b>17</b>	<b>Working with Applications.....</b>	<b>613</b>
	About applications.....	614
	Creating an application .....	616
	Reusing an application .....	622
	Running an application.....	624
	Identifying your application.....	625
	Changing the application icon on Macintosh.....	626
	Running a report, form, or pipeline.....	627
	Managing the toolbar.....	628
	Managing the open reports, forms, and pipelines .....	628
	Using the query governor in an application .....	628
	Using a pipeline in an application.....	630
	Executing pipelines .....	630
	Modifying the pipeline object's definition .....	633
<b>18</b>	<b>Deploying Your Application on Windows .....</b>	<b>637</b>
	About deploying applications .....	638
	Deploying an application .....	639
	Installing the Deployment Kit on a user's machine .....	640
	Completing the deployment process.....	642
	Making the data source available.....	642
	Configuring an ODBC driver.....	642
	Modifying the application's initialization file .....	645
	Deploying OCX controls .....	646
	Starting the deployed application .....	648
	The deployed application .....	649
	Deploying the SQL Anywhere Desktop Runtime System .....	655
	Making the data source available.....	655
	Configuring SQL Anywhere's ODBC driver and data sources .....	656
	The deployed SQL Anywhere runtime files .....	657

<b>19</b>	<b>Deploying Your Application on the Macintosh .....</b>	<b>659</b>
	About deploying applications .....	660
	Basic steps for deploying an application .....	661
	Installing the Deployment Kit on a user's machine .....	662
	Completing the deployment process .....	663
	Making the data source available .....	663
	Configuring an ODBC driver .....	663
	Installing application files .....	665
	The deployed application .....	666
 <b>20</b>	 <b>Working in a Multiplatform Environment.....</b>	 <b>669</b>
	Your working environment .....	670
	Sharing files in a multiplatform environment .....	671
	Object libraries .....	671
	Text files .....	672
	Initialization files .....	673
	Image files .....	674
	SQL Anywhere databases.....	674
	Applications for Macintosh and Windows.....	676
	Saving text files—what line endings to use.....	677
	Designing multiplatform reports and forms .....	678
	Placing objects in reports and forms .....	678
	Using fonts .....	678
	Screen resolutions and pixel size .....	683
	Platform support for InfoMaker features.....	684
 <b>PART 6</b>	 <b>REFERENCE</b>	
 <b>21</b>	 <b>Identifiers.....</b>	 <b>687</b>
	Rules .....	688
	Reserved words .....	689
 <b>22</b>	 <b>Operators and Expressions .....</b>	 <b>691</b>
	Where you use expressions .....	692
	Operators used in expressions .....	695
	Arithmetic operators .....	695
	Relational operators .....	696
	Logical operators .....	698
	Concatenation operator .....	700
	Operator precedence in expressions .....	701
	Overriding the precedence order.....	701
	Matching text patterns .....	702

<b>DataWindow Painter and InfoMaker Functions .....</b>	<b>705</b>
Using DataWindow painter and InfoMaker functions .....	706
Four examples .....	708
Example 1: counting NULL values in a column .....	708
Example 2: counting male and female employees .....	710
Example 3: creating a row indicator .....	713
Example 4: displaying all data when a column allows NULLs .....	715
Other examples .....	717
Abs .....	718
Asc .....	719
Avg .....	720
Bitmap .....	723
Case .....	724
Ceiling .....	726
Char .....	727
Cos .....	728
Count .....	729
CrosstabAvg .....	731
CrosstabCount .....	735
CrosstabMax .....	737
CrosstabMin .....	739
CrosstabSum .....	741
CumulativePercent .....	743
CumulativeSum .....	745
CurrentRow .....	747
Date .....	748
DateTime .....	749
Day .....	750
DayName .....	751
DayNumber .....	752
DaysAfter .....	753
Describe .....	754
Exp .....	755
Fact .....	756
Fill .....	757
First .....	758
GetRow .....	760
GetText .....	761
Hour .....	762
If .....	763
Int .....	764
Integer .....	765
IsDate .....	766



IsNull .....	767
IsNumber.....	768
IsRowModified.....	769
IsRowNew .....	770
IsSelected .....	771
IsTime .....	772
Large .....	773
Last .....	776
Left .....	778
LeftTrim .....	779
Len .....	780
Log .....	781
LogTen .....	782
Long .....	783
LookUpDisplay .....	784
Lower .....	785
Match .....	786
Max .....	789
Median .....	791
Mid .....	794
Min .....	795
Minute .....	797
Mod .....	798
Mode .....	799
Month .....	802
Now .....	803
Number .....	804
Page .....	805
PageAcross .....	806
PageCount .....	807
PageCountAcross .....	808
Percent .....	809
Pi .....	812
Pos .....	813
ProfileInt .....	814
ProfileString.....	816
Rand.....	818
Real.....	819
RelativeDate.....	820
RelativeTime .....	821
Replace .....	822
RGB .....	823
Right.....	825
RightTrim.....	826

Round.....	827
RowCount .....	828
RowHeight.....	829
Second.....	830
SecondsAfter.....	831
Sign .....	832
Sin .....	833
Small .....	834
Space .....	837
Sqrt.....	838
StDev .....	839
StDevP .....	842
String.....	845
Sum.....	847
Tan .....	849
Time .....	850
Today .....	851
Trim .....	852
Truncate .....	853
Upper .....	854
Var.....	855
VarP .....	858
WordCap.....	861
Year.....	862

**APPENDIX**

<b>The Powersoft Repository .....</b>	<b>865</b>
About the Powersoft repository .....	866
The PBCatTbl table .....	867
The PBCatCol table .....	869
The PBCatFmt table.....	870
The PBCatVld table.....	871
The PBCatEdt table .....	872
Available edit style types .....	872

# About This Book

## Subject

This book describes InfoMaker, what you use it for, and how you work in its environment to accomplish your goals. The book shows you how to use InfoMaker to create reports, queries, forms, data pipelines, and applications. It also shows you how to work with databases in InfoMaker.

## Audience

This book is for anyone who is using InfoMaker to work with data. Although the book does not assume you have knowledge about any particular topic, having some familiarity with relational databases and SQL is helpful. Consult books on these topics as needed.

---

### InfoMaker works with many DBMSs

This book describes how to use InfoMaker using a SQL Anywhere database for examples. You use InfoMaker with many different DBMSs, as described in *Connecting to Your Database*.

---

## Where to find information

InfoMaker is configurable at installation, so tools installed vary. To help you do your work more easily, the book is divided into parts that focus on accomplishing particular goals:

<b>If you want to</b>	<b>Use these parts of the book</b>
Learn about the environment and work with databases	Part One, The InfoMaker Environment
Create reports	Part Two, Reports
Create queries	Chapter 4, Defining Reports
Create data pipelines	Part Three, Data Pipelines
Create forms	Part Four, Forms
Create applications	Part Five, Applications

**If you want to**

Look up information about identifiers, operators, expressions, InfoMaker functions, and the Powersoft repository

**Use these parts of the book**

Part Six, Reference  
The Appendix, The Powersoft Repository

# The InfoMaker Environment

This part introduces you to InfoMaker and describes how to work in and manage its environment. It also describes how to work with databases, tables, views, and extended attributes.

---

## **Access to the Database painter**

To have access to the Database painter, install InfoMaker's database component.

---



# Working with InfoMaker

About this chapter

This chapter describes the basics of working with InfoMaker and its painters.

Contents

<b>Topic</b>	<b>Page</b>
What is InfoMaker?	4
What you can do using InfoMaker	7
The InfoMaker environment	14
Painter summary	16
Using online Help	18
Using popup menus	20
Using property sheets	22
Using toolbars	24
Using the file editor	33
Changing fonts	37
Defining colors	38
Managing the initialization file	39
Starting InfoMaker from the command line	42

## What is InfoMaker?

InfoMaker is a reporting tool

InfoMaker is a powerful and easy-to-use reporting tool that lets you query databases and create sophisticated and effective custom reports of data.

But InfoMaker is much more than a reporting tool. When optional painters are installed, it's a tool that also lets you work with data in a database.

InfoMaker is a personal data assistant

InfoMaker lets you work with data in many ways—always with no programming required.

InfoMaker provides built-in connectivity to a broad range of desktop and server-based databases. Some versions of InfoMaker also include the powerful SQL Anywhere database management system (DBMS) that enables you to create your own databases and use the built-in Powersoft Demo Database (a Sybase SQL Anywhere database) to create reports and other InfoMaker objects.

**FOR INFO** For information about supported DBMSs, see *Connecting to Your Database*.

When you work in InfoMaker, you work in a graphical environment—and working with data in this environment means you don't need to understand SQL, the standard programming language for talking to databases. InfoMaker creates all SQL statements behind the scenes as you build your reports and other objects graphically.

What you create in InfoMaker

In InfoMaker, you can create the following objects:

- ◆ Reports to view data
- ◆ Forms to view and change data
- ◆ Queries to automatically retrieve data for reports or forms
- ◆ Pipelines to pipe data from one database (or DBMS) to another
- ◆ Applications to bundle reports and forms and distribute them to users

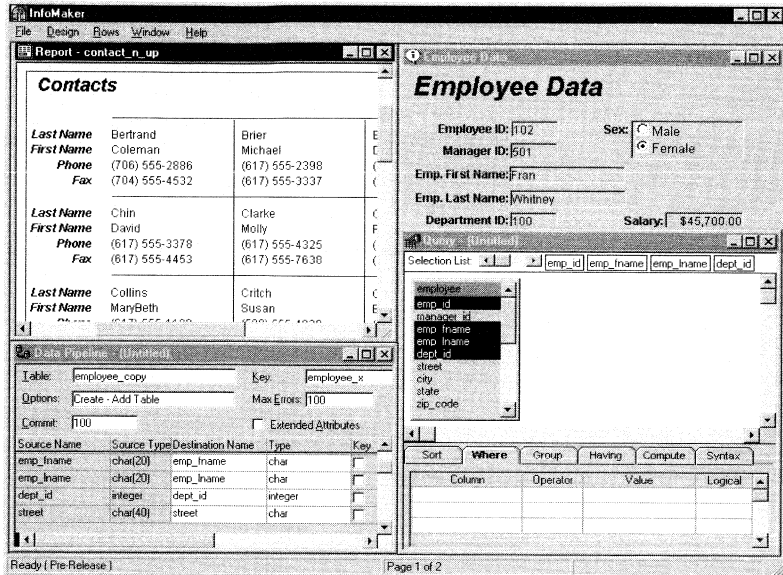
You can see many examples of the objects you create in InfoMaker in this chapter.

Painters

In InfoMaker, you do your work in **painters**. A painter is a tool you use to create and work with objects of a particular type. For example, in the Report painter, you create and work with reports, and in the Data Pipeline painter, you create and work with data pipelines. Painters provide you with an assortment of tools (some of which are additional painters) for working with InfoMaker objects.



Here's InfoMaker with the Report painter, the Form painter, the Data Pipeline painter, and the Query painter open at the same time. The painter windows have been tiled and resized to show all the painters:



## Customizing InfoMaker at installation

When you install InfoMaker, you choose various combinations of painters to install. You can add or remove any component, except for the base, at any time. The base combination includes the Report painter, the Query painter, and the Environment painter. Additional optional painters include the Database painter, the Table painter, the Form painter, and the Data Pipeline painter. You (or others in your organization who install and set up your software) can customize the InfoMaker environment at installation to meet your needs.

To work with these objects in InfoMaker	Install these painters
Reports	Report Query Environment
Databases	Database Table

<b>To work with these objects in InfoMaker</b>	<b>Install these painters</b>
Forms	Form
Data pipelines	Data Pipeline

FOR INFO For complete information about installing InfoMaker, see the *Installation Guide*.

## What you can do using InfoMaker

You use InfoMaker to create reports, queries, forms, data pipelines, and applications. You can also work with databases.

### Reports

In InfoMaker, you use the Report painter to create sophisticated reports of data. You can easily group and summarize data. You can view reports on the screen or print them. You cannot change data in a report. To change data, you use the Database painter or the Form painter.

---

**The Report Painter is always available**

The Report painter is part of InfoMaker's base component.

---

#### Types of reports

InfoMaker provides a variety of report styles:

Composite	Grid	OLE 2.0 (Windows only)
Crosstab	Group	RichText
Freeform	Label	Tabular
Graph	N-Up	

Here are a few sample reports:


Freeform report


Information about my contacts	
Last Name: <i>Bertrand</i> First Name: <i>Colleen</i> Job: <i>Documentation</i> Street: <i>78 Dunster Pl.</i> City: <i>Schaumburg</i> State: <i>IL</i> Zip: <i>60173</i> Phone: <i>(708)555-2896</i> Fax: <i>(708)555-4532</i>	<p style="text-align: center;"><i>Personal notes</i></p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
Last Name: <i>Brier</i> First Name: <i>Michael</i> Job: <i>Administration</i> Street: <i>55 Blackstone St.</i> City: <i>Arlington</i> State: <i>MA</i> Zip: <i>02174</i> Phone: <i>(617)555-2398</i> Fax: <i>(617)555-3337</i>	<p style="text-align: center;"><i>Personal notes</i></p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

Label report

 Jane Hildebrand Marketing 1280 Washington St. Emeryville, MI 94608	 Larry Simmon Sales 34 Granville St. Houston, TX 77079	 Sus Proc 45 C Dan
 Terry Lambert Administration 204 Page St. Canton, MA 94608	 Dorothy Sullivan Customer support 54 Minuteman Dr. Lincoln, MA 01742	 Ros Fine 78 E Mar
 Beth Glassmann Product development 44 Oak St. Lexington, MA 02173	 Gene Powell Training 552 West Main St. Lexington, MA 02173	 Jeff Mar 68 F Lex
 Molly Clarke Sales 55 Pine Grove Rd.	 William Kelley Documentation 16 Rainbow Rd.	 Tho Cus 64 E

Group report

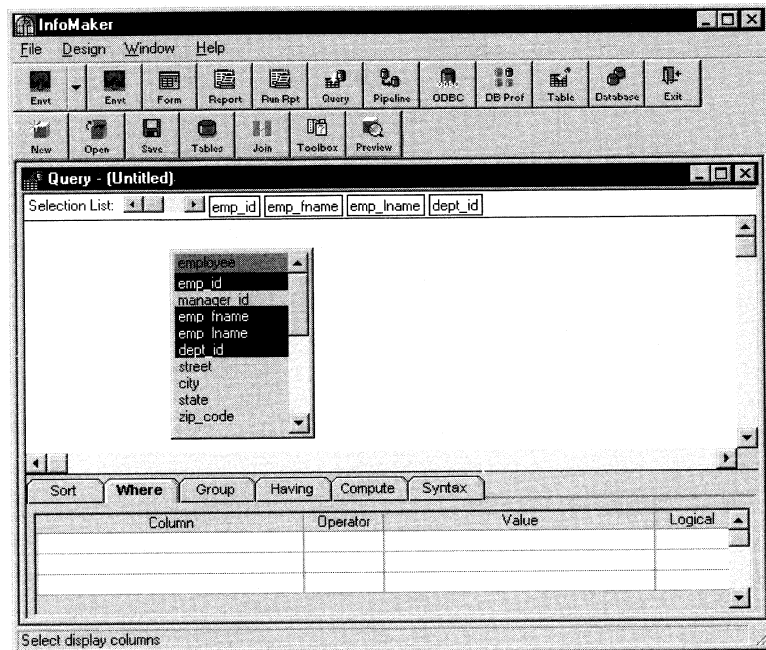
		Employee Information			page 1
	First Name	Last Name	City	Salary	
<b>Department:</b>	Software Development				
105	Alan	Chamberlain	Plymouth	\$35,000	
247	John	Spellman	Waltham	\$43,610	
300	Debbie	O'Connor	Menlo Park	\$37,900	
318	Peter	Ciccone	Milton	\$41,701	
479	Jo Marie	Houston	Pembroke	\$39,876	
1090	Bill	Smith	Acton	\$51,411	
2133	Cathy	Tyler	Vineyard	\$32,550	
2145	Bert	Simpson	Boston	\$56,220	
3400	Craig	James	Milton	\$90,500	
<i>Number of employees:</i> 9 <i>Average salary:</i> \$47,641 <i>Total salary:</i> \$428,768					
<b>Department:</b>	Business Services				
667	Ronald	Garcia	Abington	\$52,000	
703	Michael	Stanley	Westwood	\$41,501	
855	Richard	McMahon	Groton	\$24,892	
902	Edward	Fitzgerald	Gloucester	\$77,500	
2100	James	Tyler	Nantucket	\$45,200	
<i>Number of employees:</i> 5 <i>Average salary:</i> \$48,219 <i>Total salary:</i> \$241,093					

		Employee Information			page 3
	First Name	Last Name	City	Salary	
<b>Department:</b>	Marketing				
576	Thomas	Sinclair	Plymouth	\$60,000	
<i>Number of employees:</i> 1 <i>Average salary:</i> \$60,000 <i>Total salary:</i> \$60,000					
<hr/> <i>Total number of employees:</i> 21 <i>Lowest salary:</i> \$24,892 <i>Highest salary:</i> \$90,500 <i>Average salary:</i> \$51,239					
<table border="1" style="margin: auto;"> <tr> <td><b>Total salary:</b> \$1,076,026</td> </tr> </table>					<b>Total salary:</b> \$1,076,026
<b>Total salary:</b> \$1,076,026					

Queries

Reports and forms both use data from your database. In InfoMaker, you use the Query painter to define queries that specify your data requirements. When you want to create a new report or form using that data, you can simply use the query as the source of your data, without redefining the data.

Here is a sample query:



---

**The Query painter is always available**

The Query painter is part of InfoMaker's base component.

---

## Forms

In InfoMaker, you use the Form painter to create and run interactive forms to view and change data. InfoMaker provides four form styles:

Freeform

Grid

Master/Detail One-To-Many

Master/Detail Many-To-One

PowerBuilder developers in your organization can create custom form styles for you to use.

Here are some sample forms:

Freeform form

**Employee Data**

**Employee ID:** 102 **Birth Date:** 06/05/1958

**Manager ID:** 501 **Soc. Sec. No.:** 017-34-9033

**Emp. First Name:** Fran **Salary:** \$45,700.00

**Emp. Last Name:** Whitney **Start Date:** 02/26/1986

**Department ID:** 100 **Termination Date:** 00/00/0000

**Street:** 49 East Washington Street **Status:**  Active  
 Terminated  
 On Leave

**City:** Needham

**State:** MA

**Zip Code:** 02192 **Health Insurance:**

**Phone:** (617) 555-3985 **Life Insurance:**

**Sex:**  Male **Day Care:**   
 Female

Master/detail many-to-one form

**Customer Details**

Company Name	Customer ID
The Power Group	101
AMF Corp.	102
Darling Associates	103
P.S.C.	104

**Customer:** The Power Group 101

**Address:** 3114 Pioneer Avenue

**City:** Rutherford

**State:** NJ

**Zip Code:** 07070-

**Phone Number:** (201) 555-8988

**Contact:** Michaels Devlin

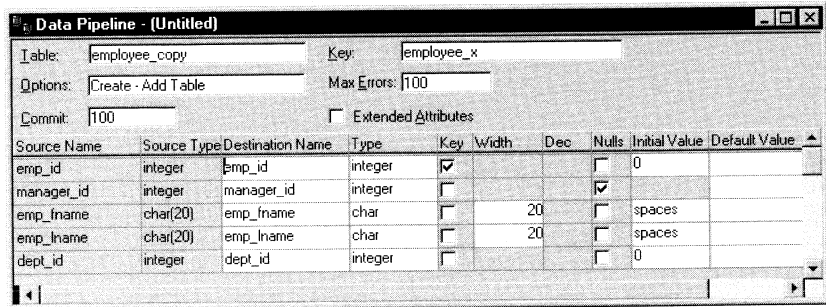
**Access to the Form painter**

You can create forms only if you install the Form painter.

## Data pipelines

In InfoMaker, you use the Data Pipeline painter to create and execute data pipeline definitions to pipe data from one or more source tables to a new or existing destination table.

Here is a sample data pipeline:



---

### Access to the Data Pipeline painter

You can create data pipelines only if you install the Data Pipeline painter.

---

## Applications

You can use your reports, forms, and data pipelines within the InfoMaker environment; you can also bundle them in a fully functional database-maintenance and reporting application that can be used outside the InfoMaker environment.

You create an application by using the Environment painter to create an executable file. You can use the application yourself; you can also distribute the executable file and some additional files to other users, who can then run the reports and forms in your application with aliases or shortcuts.

---

### The Environment painter is always available

The Environment painter is part of InfoMaker's base component.

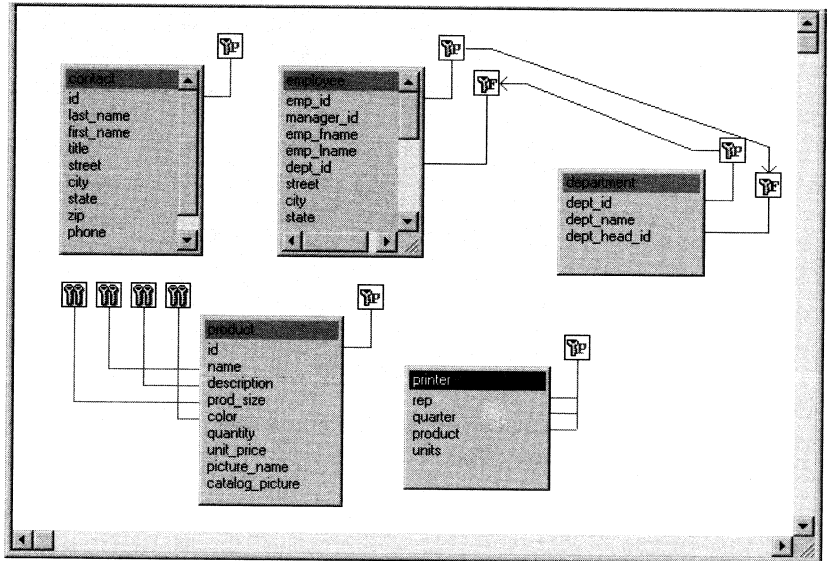
---



## Database management

The data you are working with is stored in a database. In InfoMaker, you use the Database painter to work with databases and administer them. In a database, you can create tables (which hold the data), views (which provide an easy way to use the data), indexes, and keys.

The Database painter provides a graphical interface that helps you work with databases as shown in this illustration:



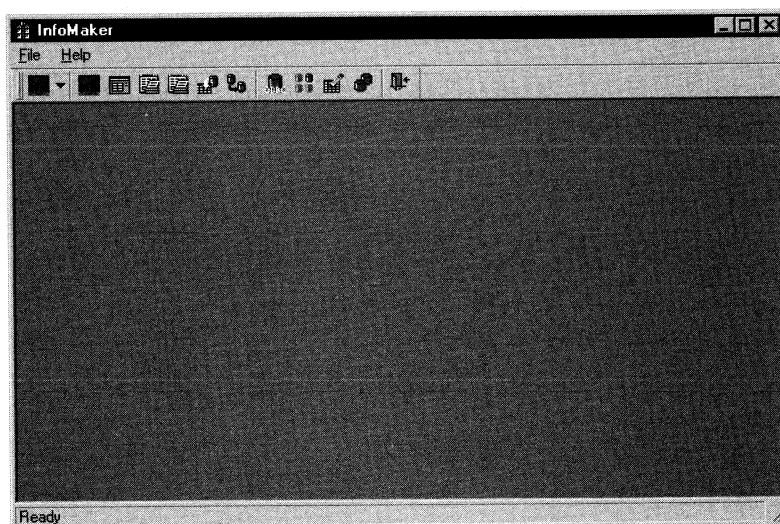
You can also define extended attributes for columns in tables. These extended attributes let you store information about columns in the database for use in reports and forms. For example, you can define an edit style and a validation rule for a column. Once they are defined, anytime you use that column in a form, each entry in the column is checked against the validation rule. If the entry doesn't pass validation, InfoMaker tells you.

### Access to the Database painter

You can work with databases only if you install the Database painter.

## The InfoMaker environment

When you start InfoMaker, it opens in a window that contains a menu bar and the PowerBar:



You can open painters and perform other tasks by clicking buttons in the PowerBar.

### About the PowerBar

The PowerBar displays when you begin an InfoMaker session. The PowerBar is the main control point for working in InfoMaker. From the PowerBar you can open an InfoMaker painter or customize InfoMaker to meet your needs.

The set of buttons you see in your PowerBar is dependent on which painters are installed. Buttons for the Form painter, Data Pipeline painter, Table painter, and Database painter are not in the PowerBar if these painters are not installed.

#### Customizing the PowerBar

You can customize the PowerBar. For example, you can choose whether to display text in the buttons, move the PowerBar around, and add buttons for operations you perform frequently.

FOR INFO For more information, see "Using toolbars" on page 24.

## About PowerTips

By default, InfoMaker displays a brief description of the button, called a **PowerTip**, when you leave the mouse pointer over a button for a second or two.

## About the PowerPanel

Like the PowerBar, the PowerPanel enables you to open painters and tools and perform other activities. It contains all the tools that are available throughout InfoMaker, including tools that are not in the default PowerBar.

### ❖ To use the PowerPanel:

- 1 Select File>PowerPanel from the menu bar.

The PowerPanel displays.

- 2 Click an item on the list to access a painter or tool and click OK.

---

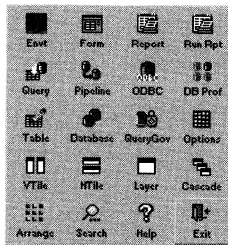
### Shortcut

To access a PowerPanel item quickly, type the first letter or two of the item name. The PowerPanel jumps to that item immediately. For example, to jump to the System Options item, type S.

---

### About the PowerPanel dropdown toolbar

The PowerPanel dropdown toolbar offers a quick way to access most PowerPanel items. When InfoMaker first opens, the two buttons at the left of the PowerBar both open the Environment painter. The button on the left has a down arrow next to it. Click the arrow to display the PowerPanel dropdown toolbar:



Whenever you click a button in either the PowerBar or the PowerPanel dropdown toolbar, that button replaces the leftmost button in the PowerBar.

## Painter summary

The buttons in the PowerBar and PowerPanel represent each of the main painters and tools frequently used in InfoMaker:

Painter or tool	What you do
Environment	Manage your InfoMaker environment and create executable versions of forms and reports
Report painter	Build and preview reports of data in your database
Run Report	Preview reports of data in your database
Query painter	Graphically define and save SQL SELECT statements for reuse with forms and reports
Data Pipeline painter	Transfer data from one data source to another
Configure ODBC	Define a data source that uses ODBC
Database Profiles	Define and use named sets of parameters to connect to a particular database
Table painter	Create database tables, alter existing tables, and define keys, indexes, and relationships between tables
Database painter	Maintain databases, control user access to databases, and manipulate data in databases
File Editor	Edit text files such as source, resource, and initialization files
Query governor	Set data selection and retrieval preferences
System options	Set InfoMaker preferences such as initialization path, fonts, and database connection preferences
Help	Invoke the InfoMaker online Help system to give you quick answers to questions

## Opening a painter or tool

There are several ways to open a painter or tool:

From here	You can
PowerBar	Click the button for the painter or tool
PowerPanel	Select the painter or tool

<b>From here</b>	<b>You can</b>
PowerPanel dropdown toolbar	Click the button for the painter or tool
Environment painter	Double-click an object to open it and start the corresponding painter
File menu	Select one of the last four objects you've worked on—they are listed at the bottom of the File menu
Anywhere	Use shortcut keys to open a painter or tool directly—shortcut keys are listed in the PowerPanel

## Using online Help

InfoMaker has online Help that provides both reference and task-oriented information.

How to access Help

You can get Help in any of these ways:

Approach	What it does
Use the Help menu on the menu bar	Displays the Help contents, the Welcome to InfoMaker window, or Help for the current painter
Click the Help button on the PowerPanel (you can add the Help button to the PowerBar if you want to)	Displays the Help contents
Press F1	Displays the Help contents
Click the Help button in a dialog box	Displays information about that dialog box

Learning to use online Help

To get information on using Help, press F1 anywhere within online Help.

Using the popup menu

InfoMaker online Help provides a popup menu with shortcuts to features available on the Help menu bar.

To display the popup menu in online Help, click the right mouse button on Windows, and press the OPTION key and click the mouse button on Macintosh.

## Linking to the online books

Some Help topics provide links you can click to go to the Powersoft Online Books. These links are represented by a picture of a CD-ROM.

FOR INFO For information about installing the Powersoft Online Books and reading them from a CD-ROM or from a server, see the *Installation Guide*.

FOR INFO For information about searching and using the annotation features in the Powersoft Online Books, see the *Using the Powersoft Online Books* collection in the Powersoft Online Books.

Online books on the Macintosh

Powersoft Online Books on the Macintosh are only available through an external folder.

❖ **To view online books on the Macintosh:**

- 1 Open the folder where you installed the Powersoft Online Books.
- 2 Open the Online Book folder.
- 3 Double-click the InfoMaker Online Book alias.  
DynaText opens and displays the top-level collection.

## Using popup menus

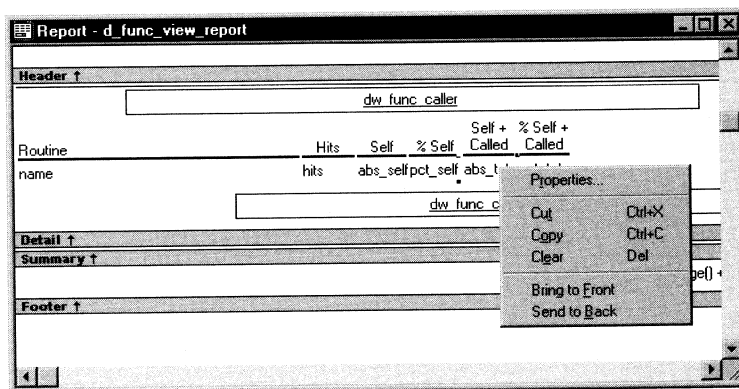
InfoMaker provides a context-sensitive popup menu that lists:

- ◆ Actions appropriate to the currently selected object or the current position of the pointer
- ◆ Where appropriate, a Properties menu item for accessing the property sheet associated with the current object or the current position of the pointer

The popup menu is available almost everywhere in InfoMaker.

### Example

For example, the following screen shows the popup menu for a column in a report:



### ❖ To display a popup menu:

- 1 Select an object, or position the pointer on an object or in an open area of the workspace.
- 2 Click the right mouse button.

---

#### On Macintosh

On the Macintosh, press the CONTROL key while you click the mouse.

---

The appropriate popup menu displays.

- 3 Click the menu option you want.

### Working with multiple objects

If you want to perform an action on multiple objects, you can select them (you can use lasso selection to do this) and display the popup menu.



When multiple objects are selected, you *cannot* use Properties from the popup menu to change their characteristics. The Properties menu item only works on single objects. You *can* use a painter's StyleBar to change properties for multiple objects.

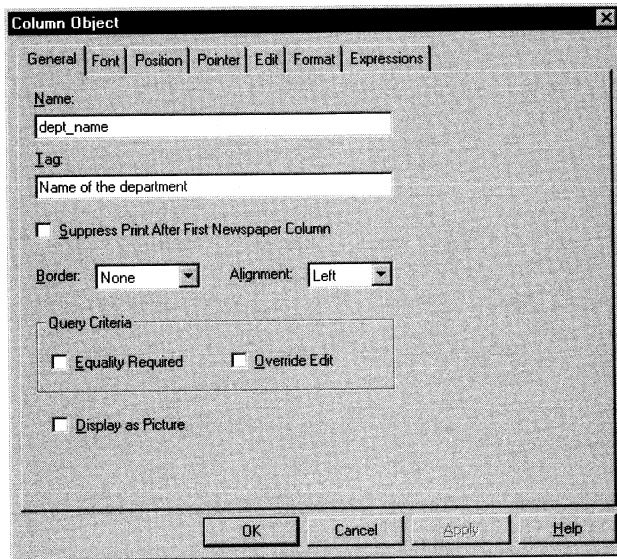
**FOR INFO** For more information about working with multiple objects, see "Selecting objects in the Report painter" on page 200 and "Selecting objects in the Form painter" on page 570.

## Using property sheets

A **property sheet** is a tab dialog box you use to set properties associated with an object, painter, or tool by making changes in one or more tabs in the dialog box.

### Example

For example, for a column in a report, you can set several different kinds of properties (general, font, position, pointer, edit style, display format, and conditional expressions) by clicking appropriate tabs in the Column Object property sheet:



### Property sheet buttons

Each property sheet has OK, Cancel, Apply, and Help buttons. The Apply button is enabled when you make a change on one tab:

Use this button	To do this
OK	Apply the properties you've set on all tabs and close the property sheet
Cancel	Close the window and apply no new changes
Apply	Apply the properties you've set on all tabs immediately without closing the property sheet
Help	Get Help on setting properties for the tab that displays

**Displaying property sheets**

You can display property sheets in four ways (the first two are usually faster):

- ◆ By selecting Properties from the popup menu of an object, painter, or tool
- ◆ By double-clicking an object
- ◆ By selecting Object>Properties or Edit>Properties from the menu bar (depending on the painter you are working in)
- ◆ By clicking the Properties button in the PainterBar

## Using toolbars

Toolbars provide buttons for the most common tasks in InfoMaker. You can move (dock) toolbars, customize them, and create your own.

### Toolbar basics

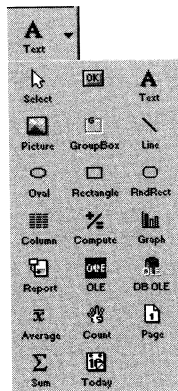
InfoMaker uses three toolbars: the PowerBar, PainterBar, and StyleBar:

<b>This toolbar</b>	<b>Has buttons for</b>	<b>And (unless hidden) displays</b>
PowerBar	Opening painters and tools	Always
PainterBar	Performing tasks in the current painter	In each painter
StyleBar	Changing the properties of text, such as font and alignment	In appropriate painters

### Dropdown toolbars

To reduce the size of toolbars, some toolbar buttons have a down arrow on the right that you can click to display a dropdown toolbar containing related buttons.

For example, the down arrow next to the Text button in the Report painter displays the Objects dropdown toolbar, which has a button for each object you can place on a report:



**Default button replaced**

The button you select from a dropdown toolbar replaces the default button on the main toolbar. For example, if you select the Picture button from the Objects dropdown toolbar, it replaces the Text button in the PainterBar.

---

## Controlling the display of toolbars

You can control:

- ◆ Whether to display individual toolbars and where
- ◆ Whether to display text on the buttons
- ◆ Whether to display PowerTips

Choosing to display text and PowerTips affects all toolbars.

**❖ To control a toolbar using the popup menu:**

- 1 Position the pointer on the toolbar and display the popup menu.
- 2 Click the items you want.

A checkmark means the item is currently selected.

**❖ To control a toolbar using the Toolbars dialog box:**

- 1 Select Window>Toolbars from the menu bar. (If no painter is open, select File>Toolbars from the menu bar.)

The Toolbars dialog box displays.

- 2 Click the toolbar you want to work with (the current toolbar is highlighted) and the options you want.

InfoMaker saves your toolbar preferences in the InfoMaker initialization file.

## Moving toolbars using the mouse

You can use the mouse to move a toolbar.

**❖ To move a toolbar with the mouse:**

- 1 Position the pointer on the grab bar at the left of the toolbar or on any vertical line separating groups of buttons.

- 2 Press and hold the left mouse button.
- 3 Drag the toolbar and drop it where you want it.

As you move, an outlined box shows how the toolbar will display when you drop it. You can line it up along any frame edge or float it in the middle of the frame.

## Docking toolbars

When you first start InfoMaker, all the toolbars display one above another at the top left of the workspace. When you move a toolbar, you can dock (position) it:

- ◆ At the top or bottom of the workspace, at any point from the left edge to the right edge
- ◆ At the left or right of the workspace, at any point from the top edge to the bottom edge
- ◆ To the left or right of, or above or below, another toolbar

### Example

These two toolbars are on separate docks:



You could customize the toolbars to reduce their size, then dock them side by side:



## Customizing toolbars

You can customize toolbars with InfoMaker buttons and with buttons that invoke other applications, such as a clock or text processor.

### Adding, moving, and deleting buttons

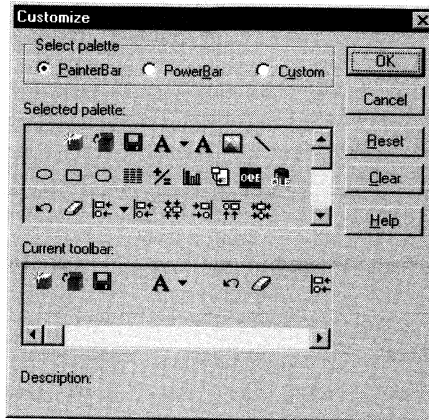
You can add, move, and delete buttons in any toolbar.

#### ❖ To add a button to a toolbar:

- 1 Position the pointer on the toolbar and display the popup menu.

- 2 Select Customize.

The Customize dialog box displays.



- 3 Click the palette of buttons you want to use in the Select palette group.
- 4 Choose a button from the Selected palette box and drag it to the position you want in the Current toolbar box.

If you choose a button from the Custom palette, another dialog box displays so you can define the button.

FOR INFO For more information, see "Adding a custom button" on page 29.

---

### Seeing what's available in the PowerBar

InfoMaker provides several buttons that do not display by default in the PowerBar, but which you can add. To see what is available, scroll the list of buttons and select one. InfoMaker lists the description for the selected button.

---

#### ❖ To move a button on a toolbar:

- 1 Position the pointer on the toolbar, display the popup menu, and select Customize.
- 2 In the Current toolbar box, select the button and drag it to its new position.

#### ❖ To delete a button from a toolbar:

- 1 Position the pointer on the toolbar, display the popup menu, and select Customize.

- 2 In the Current toolbar box, select the button and drag it outside the Current toolbar box.

## Resetting a toolbar

You can restore the original setup of buttons on a toolbar at any time.

❖ **To reset a toolbar:**

- 1 Position the pointer on the toolbar, display the popup menu, and select Customize.
- 2 Click the Reset button, then Yes to confirm, then OK.

## Clearing or deleting a toolbar

Whenever you want, you can remove all buttons from a toolbar. If you don't add new buttons to the empty toolbar, the toolbar is deleted. You can delete both built-in toolbars and toolbars you've created.

---

**To recreate a toolbar**

If you delete one of InfoMaker's built-in toolbars, you can recreate it easily. For example, to recreate the PowerBar, select PowerBar1 in the New Toolbar dialog box.

**FOR INFO** For information about creating new toolbars and about the meaning of PowerBar1, see "Creating new toolbars" on page 31.

---

❖ **To clear or delete a toolbar:**

- 1 Position the pointer on the toolbar, display the popup menu, and select Customize.
- 2 Click the Clear button, then Yes to confirm.  
The Current toolbar box in the Customize dialog box is emptied.
- 3 Select new buttons for the current toolbar and click OK.  
*or*  
Click OK to delete the toolbar.



## Adding a custom button

You can add a custom button to a toolbar. A custom button can:

- ◆ Invoke an InfoMaker menu item
- ◆ Run an executable (application) outside InfoMaker
- ◆ Run a query or report
- ◆ Assign a display format or create a computed field in a report

### ❖ To add a custom button:

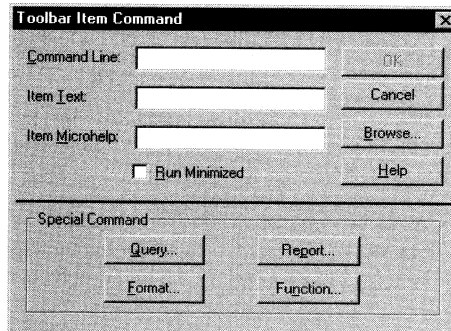
- 1 Position the pointer on the toolbar, display the popup menu, and select Customize.

- 2 Select Custom in the Select Palette group.

The custom buttons display in the Selected Palette box.

- 3 Select a custom button and drag it to where you want it in the Current toolbar box.

The Toolbar Item Command dialog box displays. Different buttons display in the dialog box depending on which toolbar you are customizing:



- 4 Fill in the dialog box as follows:

To have the button	Do this
Invoke an InfoMaker menu item	<p>Type <b>@MenuBarItem.MenuItem</b> in the Command Line box. For example, to have the button mimic the Open item on the File menu, type <b>@File.Open</b></p> <p>You can also use a number to refer to a menu item. The first item in a dropdown/cascading menu is 1, the second item is 2, and so on. Separator lines in the menu count as items. Example: <b>@Edit.Align Controls.5</b></p>
Run an executable outside InfoMaker	<p>Type the name of the executable in the Command Line box. Specify the full path name if the executable is not in the current search path</p> <p>To search for the filename, click the Browse button</p>
Run a query	Click the Query button and select the query from the displayed list
Run a report	Click the Report button and select a report from the displayed list. You can then specify command-line arguments in the Command Line box, as described below
Assign a display format to a column in a report	<p>(Report painter only) Click the Format button to display the Display Formats dialog box. Select a data type, then choose an existing display format from the list or define your own in the Format box</p> <p>FOR INFO For more about specifying display formats, see Chapter 6, "Displaying and Validating Data"</p>
Create a computed field in a report	(Report painter only) Click the Function button to display the Function for Toolbar dialog box. Select the function from the list

- In the Item Text box, specify the text associated with the button in two parts separated by a comma: the text that displays on the button and text for the button's PowerTip:

*ButtonText, PowerTip*

For example:

*Save, Save File*

If you specify only one piece of text, it is used for both the button text and the PowerTip.

- In the Item MicroHelp box, specify the text to appear as MicroHelp when the pointer is on the button.

## Supplying arguments with reports

If you defined the button to run a report, you can specify arguments in the command line in the Toolbar Item Command dialog box.

Argument	Meaning
<i>/l LibraryName</i>	Specifies the library containing the report
<i>/o ReportName</i>	Specifies the report
<i>/r</i>	Runs the report
<i>/ro</i>	Runs the report but does not provide design mode for modifying the report
<i>/a "Arguments"</i>	Specifies arguments to pass to the report

The default command line is:

```
Report /o ReportName /ro
```

## Modifying a custom button

### ❖ To modify a custom button:

- 1 Position the pointer on the toolbar, display the popup menu, and select Customize.
- 2 Double-click the button in the Current toolbar box.  
The Toolbar Item Command dialog box displays.
- 3 Make your changes, as described in "Adding a custom button" on page 29.

## Creating new toolbars

InfoMaker has built-in toolbars. When you start InfoMaker, you see what is called the PowerBar. In each painter, you also see what is called the PainterBar. But *PowerBar* and *PainterBar* are actually types of toolbars you can create to make working in InfoMaker easier.

### PowerBars and PainterBars

A PowerBar is a toolbar that always displays in InfoMaker, unless you hide it. A PainterBar is a toolbar that always displays in the specific painter for which it was defined, unless you hide it:

For this toolbar type	The default is named	And you can have up to
PowerBar	PowerBar1	4 PowerBars

For this toolbar type	The default is named	And you can have up to
PainterBar	PainterBar1	8 PainterBars in each painter

Where you create them

You can create a new PowerBar anywhere in InfoMaker, but to create a new PainterBar, you must be in the workspace of the painter for which you want to define the PainterBar.

❖ **To create a new toolbar:**

- 1 Position the pointer on any toolbar, display the popup menu, and select New.

The New Toolbar dialog box displays.

---

**About the StyleBar**

In painters that don't have a StyleBar, StyleBar is on the list in the New Toolbar dialog box. You can define a toolbar with the name StyleBar, but you can only add painter-specific buttons, not style buttons, to it.

---

- 2 Select a PowerBar name or a PainterBar name and click OK.  
The Customize dialog box displays with the Current toolbar box empty.
- 3 One at a time, drag the toolbar buttons you want from the Selected palette box to the Current toolbar box.

## Deleting toolbars

FOR INFO For instructions, see "Clearing or deleting a toolbar" on page 28.

## Using the file editor

InfoMaker provides a text editor that is always available. Using the editor, you can view and modify text files (such as initialization files and tab-separated files with data) without leaving InfoMaker.

❖ **To open the file editor:**

- 1 Press SHIFT+F6 anywhere in InfoMaker.  
*or*  
Click the Edit button in the PowerBar.

---

### **Adding an Edit button**

If there is no Edit button on the PowerBar, you can add one. The button is available from the PowerBar palette.

FOR INFO For more information, see "Customizing toolbars" on page 26.

---

The File Open dialog box displays.

- 2 Open the file you want to edit.  
*or*  
Click Cancel to display an empty editing workspace.

## Setting file editing properties

The file editor has Font and Indentation properties that you can change to make files easier to read. Select Design>Options from the menu bar to open the properties sheet. If you don't change any properties, files have black text on a white background and a tab stop setting of 3 for indentation.

❖ **To specify File Editor properties:**

- 1 Select Design>Options to display the property sheet.
- 2 Choose the tab appropriate to the property you want to specify.

---

### **Editor properties apply elsewhere**

When you set properties for the file editor, the settings also apply to the Database Administration painter.

---

## Editing activities

The file editor provides a full set of basic editing facilities including:

- ◆ Opening, saving, and printing files
- ◆ Cutting, copying, pasting, and clearing selected text
- ◆ Finding and replacing text
- ◆ Undoing changes
- ◆ Commenting and uncommenting lines
- ◆ Importing and exporting text files
- ◆ Dragging and dropping text

## Using the file editor's PainterBar and menu bar

The file editor has a PainterBar that provides a shortcut for performing frequently used activities. There is also a corresponding menu item (and often a shortcut key) for each activity:

Menu item	Shortcut key	Activity
Edit>Undo	CTRL+Z	Undoes the most recent edit
Edit>Cut	CTRL+X	Cuts selected text to the clipboard
Edit>Copy	CTRL+C	Copies selected text to the clipboard
Edit>Paste	CTRL+V	Pastes the contents of the clipboard at the current cursor location; replaces any selected text
Edit>Clear	DELETE	Deletes selected text; does not place the text in the clipboard
Edit>Select All	CTRL+A	Selects all text in the workspace
Edit>Comment Selection	—	Comments out the current line or all lines containing selected text by inserting two slashes before the first character in each line
Edit>Uncomment Selection	—	Uncomments the current line or all lines containing selected text by removing the two slashes before the first character in each line

Menu item	Shortcut key	Activity
Search>Find	CTRL+F	Specifies a string for which you want to search
Search>Find Next	CTRL+G	Finds the next occurrence of the specified search string
Search>Replace	CTRL+H	Replaces the specified search string
Search>Go to Line	CTRL+SHIFT+G	Goes to a specific line number

---

**On Macintosh**

On the Macintosh, use the COMMAND key instead of the CTRL key.

---

## Dragging and dropping text

To move text, simply select it, drag it to its new location, and drop it. To copy text, press the CTRL key while you drag and drop the text.

---

**On Macintosh**

On the Macintosh, use the COMMAND key instead of the CTRL key.

---

## Executing AppleScript scripts on the Macintosh

On the Macintosh, you can execute an AppleScript script (either in source or compiled form) from within the file editor.

❖ **To open an AppleScript file for execution:**

- 1 Open the file editor.
- 2 Do one of the following:
  - ◆ Select File>Open from the menu bar and select a source file containing AppleScript.
  - ◆ Type the name of a file in the empty file editor (do not enclose the filename in quotation marks). The file can be either a source file containing AppleScript or a compiled AppleScript script.

- ◆ Create a new script by typing AppleScript commands in the file editor. You can save the new script using File>Save from the menu bar.

❖ **To execute an AppleScript script:**

- ◆ Select Edit>Execute AppleScript from the menu bar.

The contents of the file editor are executed as an AppleScript script. If you have typed the name of a file in the editor, InfoMaker executes the script in the specified file.

InfoMaker displays a dialog box with the results.



## Changing fonts

The following table summarizes the various ways you can change the fonts used in InfoMaker:

<b>For this object or painter</b>	<b>Do this</b>
A table's data, headings, and labels	In the Database or Table painter, display the table's property sheet, and change the font properties on the Data, Heading, and Label Font tabs
Objects in the Report and Form painters	Select objects and then modify settings in the StyleBar <i>or</i> Select an object, display its property sheet, and change the font properties on the Font tab
Environment painter and MicroHelp	Click the System Options button on the PowerPanel and change the font properties on the Font tab
Database Administration painter and the file editor (changes made for one of these apply to both)	Select Design>Options from the menu bar to display the editor's property sheet and change the font properties on the Font tab

Changes you make in the System Options dialog box and from the Design>Options menu selection are reflected in the InfoMaker initialization file and are used the next time you open InfoMaker.

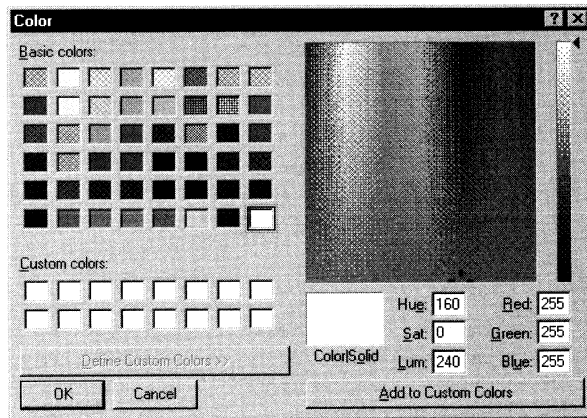
# Defining colors

You can define custom colors to use in most painters and in objects you create.

❖ **To define custom colors:**

- 1 In a painter that uses custom colors, select Design>Custom Colors from the menu bar.

The Color dialog box displays:



- 2 Define your custom colors:

Area of the Color dialog box	What you do
Basic colors	Click the basic color closest to the color you want to define to move the pointer in the color matrix and slider on the right
Custom colors palette	Modify an existing color—click a custom color, then modify the color matrix and slider. Define a new color—click an empty box, define the color, and click Add to Custom Colors
Color matrix	Click in the color matrix to pick a color
Color slider	Move the slider on the right to adjust the color's attributes
Add to Custom Colors button	After you have designed the color, click this button to add the custom color to the Custom colors palette on the left

## Managing the initialization file

When you start InfoMaker, it looks for the InfoMaker initialization file to set up your environment.

### About the initialization file

The initialization file is a text file that contains variables that specify your InfoMaker preferences. These preferences include things such as the last database you connected to, the PBL you are using, and your toolbar settings. When you perform an action in InfoMaker, InfoMaker writes your preferences to the initialization file automatically.

#### Specifying preferences

Normally, you don't need to edit the initialization file. You can specify all your preferences by taking an action, such as resizing a window or opening a new application, or by selecting Design>Options from one of the painters. But sometimes a variable doesn't appear by default in the options sheet for the painter. So if you don't see a variable whose value you want to change, use a text editor such as the InfoMaker file editor to add the variable to the appropriate section of the initialization file.

---

#### Editing the initialization file

*Do not* use a text editor to edit the InfoMaker initialization file or any preferences file while InfoMaker or your application is running. InfoMaker caches the contents of initialization files in memory and overwrites your edited InfoMaker initialization file when it exits, ignoring changes.

---

#### Format of INI files

The InfoMaker initialization file uses the Windows INI file format on all platforms. It has three types of elements:

- ◆ Section names, which are enclosed in square brackets
- ◆ Keywords, which are the names of preference settings
- ◆ Values, which are numeric or text strings, assigned as the value of the associated keyword

A variable can be listed with no value specified, in which case the default is used.

#### Default sections

The sections in the initialization file and the variables in each section can be in any order, but the variables that belong to a particular section must be in that section.

Here are some of the sections and what they contain:

Section	What it contains
Application	The name and location of the current application and library, and a history of previous applications
PB	Basic toolbar and window size preferences as well as the names of the most recently opened objects
Database	The current database profile, the list of available DBMSs, and other Database painter preferences
DBMS_Profiles	The name of the current database profile and a list of previous profiles
Profile <i>name</i>	The database profile called <i>name</i> . An additional Profile section is created for each database profile you define

Some sections are always present by default, but others are created only when you specify different preferences. The default initialization file may contain a section for specific painters: for example, the Data Window section contains preferences for the Report painter, and the SQL Painter section contains preferences for the SQL Select painter.

If you specify preferences for another painter or tool, InfoMaker creates a new section for it at the end of the file. Customized toolbar layouts are also saved in separate Toolbar sections at the end of the file.

Where the initialization file is kept

The default InfoMaker initialization file has different names and is stored in different locations on each platform:

Platform	Name	Default location of the initialization file
Windows	IM.INI	The directory where InfoMaker is installed
Macintosh	InfoMaker Preferences	System Folder:Preferences:Powersoft 6.0 Preferences

Telling InfoMaker where your initialization file is

On Windows, you can keep your initialization file in another location and tell InfoMaker where it can find it by specifying the location in the System Options tab dialog. You may want to do this if you use more than one version of InfoMaker or if you are running InfoMaker over a network.

---

#### On Macintosh

On the Macintosh, changing the initialization path in the System Options dialog box has no effect.

---

❖ **To record your initialization path:**

- 1 Open the PowerPanel and click the System Options button.

The Systems Options dialog box displays.

- 2 Enter the path of your initialization file in the Initialization Path textbox.

InfoMaker records the path in the Windows registry.

How InfoMaker finds the initialization file

On Windows, InfoMaker looks in the Windows Registry (on Windows 95 and Windows NT) or in WIN.INI (on Windows 3.1) for a path to the file, and then looks for the file in the directory where InfoMaker is installed. If InfoMaker cannot find the initialization file using the path in the Registry or WIN.INI, it clears the path value.

On the Macintosh, InfoMaker looks in System Folder:Preferences: Powersoft 6.0 Preferences.

If the initialization file is missing

If InfoMaker doesn't find the initialization file in the default location when it starts up, it recreates it. However, if you want to retain any preferences you have set, such as database profiles, keep a backup copy of your initialization file. The recreated file has the default preferences.

## Starting InfoMaker from the command line

On Windows systems, you can start InfoMaker from a command line (or the Windows 95 Run dialog box) and optionally open one of the following painters or tools:

Database painter	Form painter
Data Pipeline painter	Query painter
Environment painter	Report painter
File Editor	

To start InfoMaker and open a painter on Windows, use the following syntax:

```
{win} directory\im60.exe /P paintername
```

The use of **win** is required to start InfoMaker from the MS-DOS prompt.

### Opening an object

You can also add one or more of the following optional switches to the command line to open a specific object or create a new one:

```
{/L libraryname} {/O objectname} {/N} {/R} {/RO} {/A arguments}
```

All of these switches must follow */P paintername* on the command line, as shown in the examples after the tables.

Switch	Description
/P	Opens the specified painter
/L	Identifies the library that contains the object you want to open
/O	Identifies the object, such as a report, you want to open
/N	Creates a new report
/R	Runs the report specified with /O and allows designing
/RO	Runs the report specified with /O but does not allow designing
/A	Provides arguments for the specified report

Parameter	Description
<i>directory</i>	The fully qualified name of the directory containing InfoMaker

Parameter	Description
<i>paintername</i>	<p>The name of the painter you want to open. The default is the window that displays when you begin a new InfoMaker session</p> <p>The painter name must uniquely identify the painter. You do not have to enter the entire name. For example, you can enter <b>q</b> to open the Query painter and <b>datab</b> to open the Database painter. If you enter the full name, omit any spaces in the name (enter <b>DataPipeline</b>, for example)</p> <p>The painter name is not case sensitive. To open the file editor, you could set <i>paintername</i> to <b>FI</b> or <b>fileeditor</b></p>
<i>libraryname</i>	The name of the library that contains the object you want to open. The default is the library specified in the DefLib variable in the [PB] section of the InfoMaker initialization file
<i>objectname</i>	The name of the object you want to open

## Examples

The following examples use `im6` to represent the directory where InfoMaker is installed.

Enter this command at the Windows MS-DOS prompt to start InfoMaker and open the Database painter:

```
win im6\im60.exe /P datab
```

Enter this command in the Windows 95 Run dialog box or the Windows 3.1 File Manager Run dialog box to start InfoMaker and open the report called `d_emp_report` in the library `master.pbl`:

```
im6\im60.exe /P report /L master.pbl /O d_emp_report
```

Enter this command in the Windows 95 Run dialog box or the Windows 3.1 File Manager Run dialog box to start InfoMaker, open the report called `sum_report` in the Report painter, and run it:

```
im6\im60.exe /P report /O sum_report /R
```





# Managing Your Environment

About this chapter

This chapter describes how to manage your InfoMaker environment.

Contents

<b>Topic</b>	<b>Page</b>
About managing your environment	46
About InfoMaker libraries and special files	47
Working in the Environment painter workspace	48
Accessing objects	52
Setting the current library	53
Creating a new library	54
Copying, moving, and deleting objects in a library	55
Accessing shared queries stored on a network	58
Defining libraries for user-defined form styles	59
Query Governor	61

## About managing your environment

In InfoMaker, you use the Environment painter and the Query Governor to manage your environment. The Environment painter lets you work with the objects and libraries you create in InfoMaker. The Query Governor lets you set data selection and retrieval options.

This section describes the Environment painter.

**FOR INFO** For information about the Query Governor, see "Query Governor" on page 61.

What you can do in the Environment painter

The Environment painter workspace displays a list of objects in the current library (PBL file). In the Environment painter, you can:

- ◆ Open, copy, move, and delete objects in the current library
- ◆ Open other libraries and create new ones
- ◆ Identify query libraries stored on the network. Query libraries hold predefined query objects that can be used in creating new reports
- ◆ Identify style libraries that contain form styles developed by PowerBuilder users in your organization
- ◆ Create a runtime library that includes the objects in the current library and related resource objects
- ◆ Bundle reports and forms in an application that you and others can use

What you cannot do in the Environment painter

You cannot create new reports, forms, queries, or pipelines in the Environment painter.

## About InfoMaker libraries and special files

InfoMaker uses libraries and special files.

PBL files (InfoMaker libraries)

These are the libraries you work with in InfoMaker. In the Environment painter, you manage the objects in these libraries. In other painters, you create and access objects stored in PBLs.

---

### **When you open the Environment painter the first time**

When you open the Environment painter, you see many objects in the current library, TUTOR\_IM.PBL. This is the default InfoMaker library that contains sample objects based on the Powersoft Demo Database.

Many examples shown in this book are in TUTOR\_IM.PBL for your convenience. You can open objects in TUTOR\_IM.PBL, look at their design, and use them.

---

PSR files (Powersoft report files)

These are special InfoMaker files that contain reports. Each PSR file contains a report definition (source and object) as well as the data contained in the report when the PSR file was created.

## Working in the Environment painter workspace

❖ **To open the Environment painter:**

- ◆ Click the Environment button in the PowerBar.

The Environment painter workspace displays.

The Environment painter workspace lists the objects in the current library (PBL file), sorted by type and name. The list includes a symbol to indicate whether the object is a report, form, pipeline, or query.

---

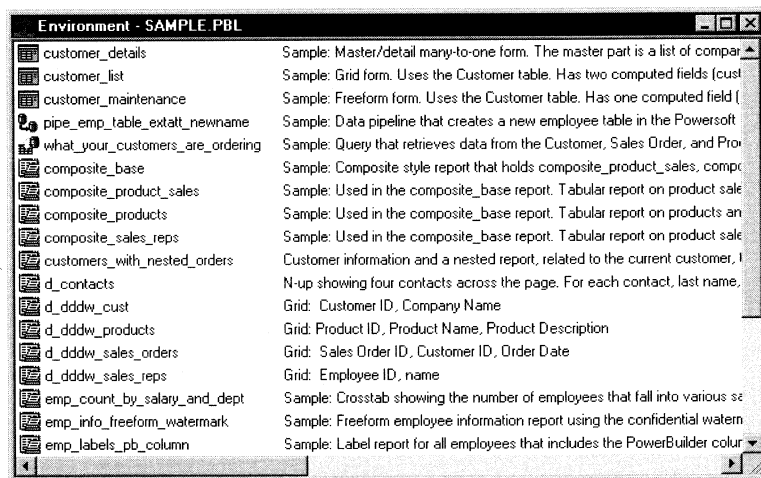
### If the Form painter and Data Pipeline painter are not installed

If the current library contains forms and pipelines, you will see them in the library list.

---

By default, if a descriptive comment was created when an object was created, the comment displays next to the name. You can hide these comments. You can also display or hide the modification date and time next to the name.

This screen shows the symbols, dates, times, and comments:



**Using the popup menu** You click the right mouse button on an object to display the popup menu, which contains items for acting on the object.

---

### **On Macintosh**

On the Macintosh, you press and hold the **COMMAND** key and the mouse button on an object to display the popup menu.

---

**Controlling what displays in the workspace** You can control whether dates/times and comments display for each object listed in the workspace. You can also edit the comment text. These procedures are described next.

## **Displaying dates and times**

You can display the last modification date and time for each object. Or you can turn off the display of dates and times.

❖ **To display dates in the Environment painter workspace:**

- ◆ Select **Design>Modification Date** from the menu bar.

InfoMaker displays the date and time when each object was last saved. A checkmark next to the **Modification Date** means dates are displayed.

## **Displaying comments**

You can display the comments for each object. Or you can turn off the display of comments.

❖ **To display comments in the Environment painter workspace:**

- ◆ Select **Design>Comments** from the menu bar.

InfoMaker displays the comments for each object. A checkmark next to **Comments** means comments are displayed.

## **Refreshing the library list**

When you save new objects or move or delete objects, the list of objects does not change unless you refresh the library list.

❖ **To refresh the library list:**

- ◆ Select Design>Refresh from the menu bar.

InfoMaker refreshes the library list and displays all objects in the current library.

## Modifying comments

You can:

- ◆ Modify existing comments for objects
- ◆ Add comments to objects that do not currently have any comments
- ◆ Modify the comments for a single object
- ◆ Modify comments for multiple objects using a shortcut

❖ **To modify comments for objects:**

- 1 Select the objects you want.

Press the CTRL key while clicking to select objects individually.

*or*

After selecting one object, press SHIFT and click another object to select all between the two selected objects.

---

**On Macintosh**

Press the COMMAND key while selecting entries individually *or* select a group of entries by selecting one entry, pressing shift, and clicking another entry to select all entries between the two.

---

- 2 Select Entry>Modify Comments from the menu bar.

InfoMaker displays the Modify Entry Comments dialog box. The information that displays is for the first object you selected. You can change existing comments, or, if there are no comments, you can enter new descriptive text.

- 3 Click OK when you have finished with the first one.

If you do not want to change the comments for an object, click Skip.

The next object displays.

- 4 Enter comments and click OK (or Skip) for each object until you have finished.

If you want to stop working on comments before you finish with the objects you selected, click Cancel. The comments you've entered until the most recent OK are retained.

The new comments display in the Environment painter workspace.

## Accessing objects

You can access objects (reports, forms, queries, and pipelines) in the Environment painter.

In the Environment painter, you can see many names, dates, and comments at the same time. This can make it easier to find objects to work on. So if you are working on existing objects, you may want to start from the Environment painter.

However, you cannot create new objects in the Environment painter. You must go to the Report painter, Form painter, Query painter, or Data Pipeline painter to do that.

## Opening an object

### ❖ To open an object in the Environment painter:

- ◆ Double-click an object in the list.

*or*

Select Edit from the object's popup menu.

InfoMaker takes you to the Report painter, Form painter, Query painter, or Data Pipeline painter workspace and opens the object. You can work on the object and save it as you work. When you close it, you return to the Environment painter.

## Previewing an object

### ❖ To preview an object in the Environment painter:

- ◆ Select Run/Preview from the object's popup menu.

InfoMaker takes you to the Report painter, Form painter, Query painter, or Data Pipeline painter and opens the object in preview. When you close the object, you first return to the design workspace and then to the Environment painter.



## Setting the current library

As you build up your collection of objects, you can keep them in one library or in different libraries in different folders. When you want to work on the objects in a particular library in a particular folder, you need to open the library in that folder in the Environment painter workspace.

---

### About libraries (PBL files)

InfoMaker uses special files called libraries to hold objects. A library is a file with the file extension PBL (pronounced *pibble*).

*filename.pbl*

---

Whenever InfoMaker is running, the current library determines which objects are listed in the Select dialog boxes in the Report painter, Form painter, Query painter, and Data Pipeline painter. When you save an object, InfoMaker puts it in the current library.

#### ❖ To set the current library:

- 1 Select File>Open from the menu bar.

InfoMaker displays the Select Library dialog box with a list of the libraries in the current folder.

- 2 Navigate to the folder and library you want and select the library.
- 3 Click Open.

InfoMaker changes the current library. The workspace updates to display the objects in the current library.

## Creating a new library

When you create a new library, the new library becomes the current library. The workspace is empty, because no objects exist in the current library.

❖ **To create a new library:**

- 1 Select File>New from the menu bar.  
InfoMaker displays the Select Library dialog box.
- 2 Navigate to the folder you want and then name the new library in the File Name box.
- 3 Click Save.

## Copying, moving, and deleting objects in a library

As you develop a collection of objects, you will probably want to reorganize them or back them up in different places. You may also want to give copies to others and to delete obsolete objects. In the Environment painter, you can copy and move objects to different libraries on different drives in different folders, and you can delete objects.

### Copying and moving objects in a library

You can copy and move objects from the current library. The difference between copying and moving is that copying results in two objects and moving results in one. For example, after you copy a report, you have two reports in two different locations. After you move a report, you still have one report, but it's in a different location.

#### ❖ To copy one or more objects in a library:

- 1 Change to the library that holds the objects you want to copy.  
FOR INFO For information, see "Setting the current library" on page 53.
- 2 Select the objects you want to copy.
- 3 Click the Copy button in the PainterBar.  
*or*  
Select Entry>Copy from the menu bar.  
InfoMaker displays the Select Library dialog box.
- 4 Navigate to the folder and library you want to copy objects to and select the library.
- 5 Click Open.

InfoMaker puts copies of the selected objects in the specified library. The originals are unchanged and remain where they were. If you make the destination library the current library, you can check to make sure your copies are where you want them.

#### ❖ To move one or more objects in a library:

- 1 Change to the library that holds the objects you want to move.  
FOR INFO For information, see "Setting the current library" on page 53.
- 2 Select the objects you want to move.

- 3 Click the Move button in the PainterBar.  
*or*  
Select Entry>Move from the menu bar.  
  
InfoMaker displays the Select Library dialog box.
- 4 Navigate to the folder and library you want to move the objects to and select the library.
- 5 Click Open.  
  
InfoMaker puts the original objects you selected in the library you specified. The originals are no longer in the current library and therefore do not display in the list in the Environment painter workspace. If you change the current library to the destination library, you can check to make sure your originals are where you want them.

## Deleting objects in a library

You can delete objects from the current library.

❖ **To delete one or more objects in a library:**

- 1 Change to the library that holds the objects you want to delete.  
  
FOR INFO For information, see "Setting the current library" on page 53.
- 2 Select the objects you want to delete.
- 3 Click the Delete button in the PainterBar.  
*or*  
Select Entry>Delete from the menu bar.  
  
InfoMaker displays a confirmation dialog box to make sure you want to delete the objects. If you have identified multiple objects, you are asked to confirm each deletion.
- 4 Click Yes or No for each confirmation.  
  
InfoMaker deletes the objects you selected and confirmed. If you confirm some but not others, only the ones you confirm are deleted. The deleted objects no longer display in the list in the Environment painter workspace.

---

**Be careful when deleting**

Once you finish confirming deletions, the objects are gone. You cannot restore them.

Remember to back up your libraries (PBLs) periodically to protect against accidental deletion.

---

## Accessing shared queries stored on a network

Your company may store a library (a PBL file) of queries on a network. Having a query library is a convenient way to make carefully developed and well-tested queries that go against corporate data available to everyone.

If you want to access a query library on a network, you use the Environment painter to identify it as a source of queries. Then when you create a new report, form, or pipeline, and specify Query as your data source, InfoMaker includes the queries from the query library in your list of available queries.

❖ **To identify a query library as a source of queries:**

- 1 Select Design>Options from the menu bar and then select the Query tab.
- 2 Enter one or more library search paths in the Library Search Path box.  
*or*  
Click Browse to select a library search path and then click Open to add the library to the Library Search Path box.

---

**On Macintosh**

On the Macintosh, enter one or more library search paths in the Library Search Path box.

You can also click Browse to select a library search path, click Add to add the library, and then click Done to add the library to the Library Search Path box.

---

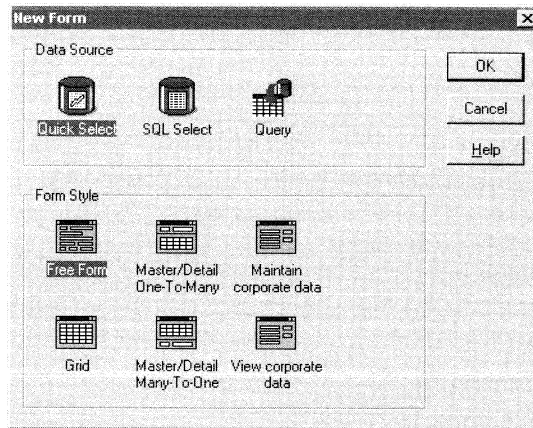
- 3 Click OK.

InfoMaker sets the library search path for queries. This means that when you start creating a new report, form, or pipeline, and choose Query as the data source, your list of available queries will include all queries in all the libraries in your search path as established in this process.

## Defining libraries for user-defined form styles

InfoMaker provides four built-in form styles: Freeform, Grid, and two Master/Detail styles. PowerBuilder users in your organization can create additional form styles that you can use to build forms.

To use the user-defined form styles, you identify the libraries (PBL files) that contain them. Then when you create a new form, the user-defined form styles display in the New Form dialog box:



### ❖ To identify a library as a source of form styles:

- 1 Select Design>Options from the menu bar.
- 2 Enter one or more library search paths in the Library Search Path box.  
*or*  
Click Browse to select a library search path and then click Open to add the library to the Library Search Path box.

---

#### On Macintosh

On the Macintosh, enter one or more library search paths in the Library Search Path box.

You can also click Browse to select a library search path, click Add to add the library, and then click Done to add the library to the Library Search Path box.

---

3 Click OK.

InfoMaker sets the library search path for user-defined form styles. When you create a new form, the form styles defined in your organization display in the Form Style box in the New Form dialog box.



## Query Governor

The Query Governor helps you manage your environment by enabling you to set data selection and retrieval options.

### Data selection options

Here are the data selection options you can set:

Data selection options	Description
Specify the maximum number of tables in a join	Specifying a maximum limits data selection. Increasing the maximum means fewer restrictions on data selection and longer retrieval times
Allow cross products	When cross products are allowed, you can have tables not joined by the join operator. One row is retrieved for each combination of rows in the tables. If table A has x rows and table B has y rows, a cross product of A and B has x times y rows, unless you specify WHERE criteria
Allow outer joins	When outer joins are allowed, all rows in a table are retrieved whether or not a matching row exists in another table
Allow SELECT DISTINCT statements	Usually a SELECT statement retrieves all rows satisfying the SELECT statement. If SELECT DISTINCT is specified, duplicate rows are not retrieved. Retrieval time is often much longer when DISTINCT is specified

### Data retrieval options

These data retrieval settings specify rows retrieved and maximum time on the client, not the server:

Data retrieval options	Description
Specify the maximum number of rows retrieved	With no maximum set, all rows are retrieved. Specifying a maximum number limits retrieval and means shorter retrieval times
Specify the maximum time for retrieval	With no maximum set, retrieval time is not limited. Specifying a maximum time limits retrieval time

---

### Using a shared InfoMaker initialization file

In some organizations, Query Governor options are specified in a shared InfoMaker Initialization file.

**FOR INFO** For information about using a shared initialization file, see *Connecting to Your Database*.

---

## Accessing the Query Governor

The Query Governor button is not in the PowerBar when you install InfoMaker.

### ❖ To access the Query Governor from the PowerBar:

- 1 Customize the PowerBar to add a Query Governor button.

**FOR INFO** For information about customizing toolbars, see "Customizing toolbars" on page 26.

- 2 Click the Query Governor button in the PowerBar.

### ❖ To access the Query Governor from the PowerPanel:

- 1 Press CTRL + SHIFT + N.  
*or*  
Select File>PowerPanel from the menu bar.

---

### On Macintosh

Press COMMAND+P *or* select File>PowerPanel from the menu bar.

---

The PowerPanel displays.

- 2 Scroll to Query Governor, select it, and then click OK.

## Using the Query Governor

When you select and retrieve data, the default Query Governor options limit you in the following ways:

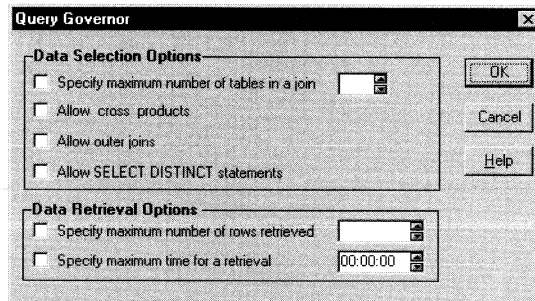
<b>You can do this</b>	<b>You can't do this</b>
Join an unlimited number of tables	Specify cross products

You can do this	You can't do this
Retrieve an unlimited number of rows	Specify outer joins
Retrieve for an unlimited time	Use SELECT DISTINCT statements

You can change the Query Governor default options anytime by selecting or deselecting the options in the Query Governor dialog box. If you are in the Select painter when you change a data selection option, the change is not enabled until you leave the Select painter and open it again.

❖ **To use the Query Governor:**

- ◆ Access the Query Governor dialog box as described above and select options:





# Managing the Database

## About this chapter

Central to your InfoMaker applications is the database. This chapter describes how you can manage the database from within InfoMaker.

## Contents

<b>Topic</b>	<b>Page</b>
About databases	66
About managing databases	69
Using the Database painter	72
Creating and deleting a SQL Anywhere database	79
Working with tables in the Database painter	81
Working with tables in the Table painter	88
Working with indexes	109
Working with keys	102
Working with views	112
Manipulating data	118
Administering the database	126

## Before you begin

You work with relational databases in InfoMaker. If you are not familiar with relational databases, you may want to consult an introductory text, such as *A Database Primer* by C. J. Date.

## About databases

A database is an electronic storage place for data. Databases are designed to ensure that data is valid and consistent, and that it can be accessed, modified, and shared.

A **database management system (DBMS)** governs the activities of a database and enforces rules that ensure data integrity. A *relational* DBMS stores and organizes data in tables.

How you work with databases in InfoMaker

You can use InfoMaker to work with the following database components:

- ◆ Tables and columns
- ◆ Keys
- ◆ Indexes
- ◆ Views
- ◆ Extended attributes

## About tables and columns

A database usually has many tables, each of which contains rows and columns of data. Each row in a table has the same columns, but a column's value for a particular row could be empty or NULL if the column's definition allows it.

Tables often have relationships with other tables. For example, in the Powersoft Demo Database that is included with InfoMaker, the Department table has a Dept\_id column, and the Employee table also has a Dept\_id column that identifies the department in which the employee works. When you work with the Department table and the Employee table, the relationship between them is specified by a **join** of the two tables.

## About keys

Relational databases use keys to ensure database integrity.

Primary keys

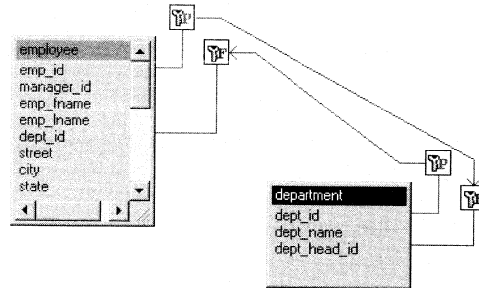
A **primary key** is a column or set of columns that uniquely identifies each row in a table. For example, two employees may have the same first and last names, but they have unique ID numbers. The Emp\_id column in the Employee table is the primary key column.

**Foreign keys**

A **foreign key** is a column or set of columns that contains primary key values from another table. For example, the Dept\_id column is the primary key column in the Department table, and a foreign key in the Employee table.

**Key icons**

In InfoMaker, columns defined as keys are displayed with **key icons** that include a P for private or F for foreign. InfoMaker automatically joins tables that have a primary/foreign key relationship, with the join on the key columns:



FOR INFO For more information, see "Working with keys" on page 102.

**About indexes**

An index is a column or set of columns you identify to improve database performance when searching for data specified by the index. You index a column that contains information you will need frequently. Primary and foreign keys are special examples of indexes.

You specify a column or set of columns with unique values as a **unique index**, represented by an icon with a single key.

You specify a column or set of columns that has values that are not unique as a **duplicate index**, represented by an icon with two keys.

FOR INFO For more information, see "Working with indexes" on page 109.

**About views**

If you often select data from the same tables and columns, you can create a **view** of the tables. You give the view a name, and each time you refer to it the associated SELECT command executes to find the data.

Views display in the Select Tables dialog box in the Database painter, but a view does not physically exist in the database in the same way that a table does. Only its definition is stored in the database, and the view is recreated whenever the definition is used.

Database administrators often create views for security purposes. For example, a view of an Employee table that is available to users who are not in Human Resources might show all columns except Salary.

FOR INFO For more information, see "Working with views" on page 112.

## About extended attributes

**Extended attributes** enable you to store information about a table's columns in special system tables called the **Powersoft repository**. Unlike tables, keys, indexes, and views, which are DBMS-specific, extended attributes are Powersoft-specific. The most powerful extended attributes determine the edit style, display format, and validation rules for the column.

FOR INFO For more information about extended attributes, see "Specifying extended attributes" on page 92. For more information about the Powersoft repository, see the Appendix, "The Powersoft Repository".



## About managing databases

InfoMaker supports many database management systems (DBMSs). For the most part, you work in the same way in InfoMaker for each DBMS. But because each DBMS provides some unique features (which InfoMaker makes use of), there are some issues that are specific to a particular DBMS.

**FOR INFO** For complete information about using your DBMS, see *Connecting to Your Database*.

## How you work with the database

You do most of your database work using the Database painter and other painters you can launch from the Database painter.

### Access to databases

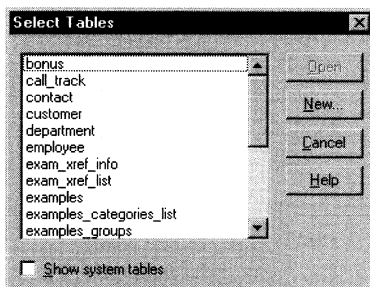
If you installed the Database painter, you can do the following in any DBMS to which you have been given access by the database administrator:

- ◆ Modify local table and column properties
- ◆ Use the Data Manipulation painter to retrieve, change, and insert data and modify existing tables
- ◆ Use the Table painter to create new local tables or modify existing tables

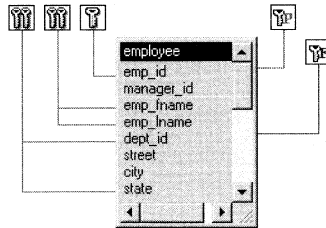
If you do not install the Database painter, you can report on data in tables to which you have access, but you cannot change data, create new tables, or modify existing tables.

### How the Database painter works

When you open the Database painter, InfoMaker displays the names of tables and views to which you have access in the current database:



You work with that database until you connect to another database. When you select tables and click Open, the Database painter graphically displays tables in the database. Indexes and keys display as icons:



In the Database painter, you can:

- ◆ Graphically display tables in the database
- ◆ Open and close tables, including tables related to a selected table
- ◆ Drop tables (remove tables from the database)
- ◆ Modify table and column properties
- ◆ Create, alter, and drop primary keys, foreign keys, and indexes
- ◆ Define and modify extended attributes for columns
- ◆ Create and drop views

---

### Creating and deleting databases

When you are connected to SQL Anywhere, you can also create a new database or delete an existing database using the Database painter.

For all other DBMSs, creating and deleting a database is an administrative task that you cannot do within InfoMaker.

---

### Related painters

From the Database painter, you can also open five related painters:

- ◆ The *Table painter*, where you create and alter tables
- ◆ The *Data Manipulation painter*, where you retrieve and manipulate data from the database
- ◆ The *Database Administration painter*, where you control access to the database and execute SQL directly
- ◆ The *View painter*, where you create views
- ◆ The *Data Pipeline painter*, where you create pipelines

The following table summarizes what you can do in each painter:

<b>Painter</b>	<b>What you can do</b>
Database	<ul style="list-style-type: none"> <li>Graphically display tables</li> <li>Open, close, and drop tables</li> <li>Modify table properties</li> <li>Modify column properties</li> <li>Create and drop keys and indexes</li> <li>Open, close, and drop views</li> <li>Define and modify extended attributes</li> </ul>
Table	<ul style="list-style-type: none"> <li>Create new tables</li> <li>Modify table properties</li> <li>Create keys and indexes</li> <li>Define and modify extended attributes</li> </ul>
Database Manipulation	<ul style="list-style-type: none"> <li>Retrieve rows</li> <li>Insert, update, and delete rows</li> <li>Save data</li> </ul>
Database Administration	<ul style="list-style-type: none"> <li>Maintain users</li> <li>Define security</li> <li>Execute SQL</li> <li>Create stored procedures</li> <li>Create triggers</li> <li>Analyze performance</li> </ul>
Data Pipeline	<ul style="list-style-type: none"> <li>Create a data pipeline definition using a selected table</li> <li>Execute the pipeline</li> </ul>
View	<ul style="list-style-type: none"> <li>Create views</li> </ul>

The Table painter, Data Pipeline painter, and Database Administration painter can all be accessed from the Database painter. But they are also standalone painters that can be opened by clicking a button in the PowerBar.

You can access the Data Manipulation painter and the View painter only from the Database painter.

## Using the Database painter

❖ **To open the Database painter:**

- 1 Click the Database button in the PainterBar.

The Select Tables dialog box displays and lists the names of all tables and views in the current database.

---

### **Suppressing the table list**

If you clear the Display Table List checkbox in the Database Preferences property sheet, InfoMaker does not display the table list when you open the Database painter. You can then use menu items as needed and click the Open button in the PainterBar to open the Select Tables dialog box.

**FOR INFO** For information about changing Database preferences, see "Modifying database preferences" on page 77.

---

- 2 Select one or more tables and click Open to display them graphically.

*or*

Click New to create a new table in the Table painter.

*or*

Click Cancel to go to the painter workspace with no tables displayed.

**FOR INFO** For more information, see "Working with tables in the Database painter" on page 81.

## Changing the database connection

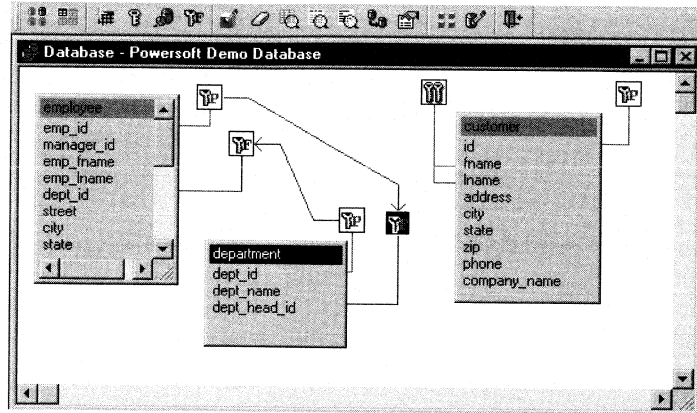
When you open a painter that communicates with the database (such as the Database painter or Report painter), InfoMaker connects you to the database you used last if you are not already connected. You can change to a different database anytime.

**FOR INFO** For more about changing the database you are connected to, see *Connecting to Your Database*.

## About the painter

Like the other InfoMaker painters, the Database painter contains a menu bar, a customizable PainterBar, and a workspace.

To work with database components, you open them in the workspace. Three Powersoft Demo Database tables are open in the workspace shown below: Employee, Department, and Customer:



InfoMaker displays table columns and icons that mark a column or a set of columns as a primary key, a foreign key, or an index. Keys and indexes were defined for the tables earlier—by you, another user, or your database administrator.

### Number of columns that display in tables

InfoMaker displays eight columns in tables in the Database painter and adds a scrollbar for larger tables. You can change the number of columns that display by setting the Columns in Table Display property in the Database Preferences property sheet.

**FOR INFO** For information about changing Database preferences, see "Modifying database preferences" on page 77.

## The menu bar and PainterBar

You can perform most common activities in a database from the Database painter's menu bar. The PainterBar provides buttons for performing all the activities listed in the following table, as well as two additional buttons for defining a database profile and opening the Database Administration painter. Use PowerTips to identify the buttons.

<b>Menu item</b>	<b>Activity</b>
Object>Select Tables	Open an existing table in the Database painter workspace or create a new table in the Table painter
Object>New>Table	Create a new table in the Table painter
Object>New>Index	Create an index for the table that's selected in the Database painter workspace
Object>New>View	Create a view in the current database
Object>New>Foreign Key	Create a foreign key for the table that's selected in the Database painter workspace
Object>Edit Object	Alter the table that's selected in the Database painter workspace using the Table painter
Object>Drop	Drop the table that's selected in the Database painter workspace
Object>Edit Data>Grid	Manipulate data in the table that's selected in the Database painter workspace using the grid format of the Data Manipulation painter
Object>Edit Data>Table	Manipulate data in the table that's selected in the Database painter workspace using the table format of the Data Manipulation painter
Object>Edit Data>Freeform	Manipulate data in the table that's selected in the Database painter workspace using the freeform format of the Data Manipulation painter
Object>Data Pipeline	Pipe data in the table that's selected in the Database painter workspace
Object>Properties	Modify the properties of the table that's selected in the Database painter workspace
Design>Arrange Tables	Arrange the tables in the Database painter workspace
File>Exit	Close the Database painter

## How the Database painter and the Table painter work together

The Database painter and the Table painter are independent painters, but when they're open at the same time they work together. For example, when you create and save a table definition in the Table painter, the Table painter notifies the Database painter that a new table exists and the new table displays in all instances of the Database painter. If you add an index for a table in the Database painter, you can see the change if you open the Table Properties property sheet for that table in the Table painter.

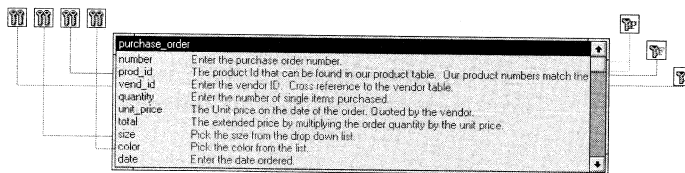
## Working with objects in the workspace

**Moving objects** You can move an object around the workspace by dragging it with the mouse. For example, to move a table, press the left mouse button on the title bar for the table and drag the table.

**Resizing objects** You can resize an object in the workspace by dragging one of its corners.

**Displaying comments, indexes, and keys** When a table is created or modified in InfoMaker, you can assign comments to the table and to each of its columns. These comments can be useful to others who will be using the tables. You can choose to display comments for tables displayed in the workspace by selecting Show Comments from the popup menu for the workspace.

Similarly, you can choose whether to display indexes and, if your DBMS supports them, primary and foreign keys:



**Using a table or column popup menu** A table open in the workspace has a table name at the top and column names below. In the illustration above, the table name is `purchase_order`.

### Table popup menu

When you point at the table name and press the right mouse button, you display the popup menu for the table.

---

**On Macintosh**

On the Macintosh, hold down the CONTROL key and press the mouse button to display popup menus.

---

From the popup menu, you can select menu items to work with the table in the Database painter or in one of the related painters:

Select this	To do this
Close	Close the table
Alter Table	Open the Table painter so that you can modify the table
Properties	Open the property sheet for the table
New	Create a new index or foreign key
Drop Table	Drop the table
Edit Data	Open the Data Manipulation painter so that you can modify the data
Data Pipeline	Open the Data Pipeline painter so that you can create a data pipeline definition with the table as the source
Print Definition	Print the table definition

**FOR INFO** For information about modifying table properties, see "Modifying table properties in the Database painter" on page 83. The Table painter and Data Manipulation painter are described later in this chapter. The Data Pipeline painter is described in Chapter 14, "Working with Data Pipelines".

**Column popup menu**

When you point at a column name and press the right mouse button, you display the popup menu for the column.

---

**On Macintosh**

On the Macintosh, hold down the CONTROL key and press the mouse button to display popup menus.

---

Select Definition to display and modify the table definition in the Table painter, and select Properties to display the column properties and modify them.

**FOR INFO** For information about using the Table painter, see "Working with tables in the Table painter" on page 88. For information about modifying column properties, see "Modifying column properties in the Database painter" on page 84.



## Keyboard alternatives

InfoMaker provides keyboard alternatives to common workspace activities:

To	Do this
Select a table or view	Press <b>TAB</b> to move left to right among the opened tables and views, or press <b>SHIFT+TAB</b> to move right to left
Display the Table painter	Select the table for which you want to display the information, then press <b>ENTER</b>
Browse an index or key	Press <b>TAB</b> or <b>SHIFT+TAB</b> to select the index or key to browse and then press <b>ENTER</b>
Scroll in the painter workspace	If you open more tables and views than can be displayed in the Database painter workspace at one time, you can scroll up or down to view all the tables and views  To scroll up, press <b>UP ARROW</b> . To scroll down, press <b>DOWN ARROW</b>

## Modifying database preferences

To modify database preferences, select **Design>Options** from the menu bar. Some of the preferences are specific to database connection. Other preferences are specific to the Database painter.

### Preferences on the General property page

The Shared Database Profiles, Use Powersoft Repository, Read Only, and Keep Connection Open preferences are database-connection specific preferences.

**FOR INFO** For information about modifying these preferences, see *Connecting to Your Database*.

The remaining preferences are specific to the Database painter.

Database preference	What InfoMaker does with the specified preference
Display Table List	InfoMaker displays the table list when you open the Database painter unless you clear the checkbox
SQL Terminator Character	InfoMaker uses the semicolon as the SQL statement terminator unless you enter a different terminator character in the textbox

<b>Database preference</b>	<b>What InfoMaker does with the specified preference</b>
Refresh Table List	When InfoMaker first displays a table list, InfoMaker retrieves the table list from the database and displays it. To save time, InfoMaker saves this list internally to use again so very large table lists don't have to be regenerated. The table list is refreshed every 30 minutes (1800 seconds) unless you specify a different refresh rate
Columns in the Table List	When InfoMaker displays tables graphically, eight table columns display unless you change the number of columns

Preferences on the Color property page

You can set colors separately for each component of the Database painter's graphical table representation: the table header, columns, indexes, primary key, and foreign keys. Set a color preference by selecting a color from a dropdown listbox.

You can design custom colors that you can use when you select color preferences. To design custom colors, select Design>Custom Colors from the menu bar and work in the Custom Colors dialog box.

## Creating and deleting a SQL Anywhere database

In InfoMaker you work within an existing database. With one exception, creating or deleting a database is an administrative task that is not performed directly in InfoMaker. The one exception is that you can create and delete a local SQL Anywhere database from within many versions of InfoMaker.

**FOR INFO** For information about creating and deleting databases, see your DBMS documentation.

### ❖ **To create a local SQL Anywhere database:**

- 1 Open the Database painter.
- 2 Select File>Create Database from the menu bar.

The Create Local Database dialog box displays.

- 3 In the Database Name box, specify the filename and path of the database you are creating.

If you do not provide a path, the database will be created in the current directory or folder. If you do not provide a file extension, the database filename will be given the extension DB.

- 4 Define other properties of the database as needed. Click the More button to define additional properties.

If you are using a non-English database, you can specify a code page in the Collation Sequence box.

**FOR INFO** For complete information about filling in the dialog box, click the Help button in the dialog box.

- 5 Click OK.

### What happens

When you click OK, InfoMaker does the following:

- ◆ Creates a database with the specified name in the specified directory or folder. If a database with the same name exists, you are asked whether you want to replace it.
- ◆ Adds a data source to the ODBC preferences file (ODBC.INI on Windows). The data source has the same name as the database unless one with the same name already exists, in which case a suffix is appended.
- ◆ Creates a database profile and adds it to the InfoMaker initialization file. The profile has the same name as the database unless one with the same name already exists, in which case a suffix is appended.
- ◆ Connects to the new database.

❖ **To delete a local SQL Anywhere database:**

- 1 Open the Database painter.
- 2 Select File>Delete Database from the menu bar.
- 3 Select the database you want to delete.  
You are prompted to confirm your action.
- 4 Click Yes to delete the database.

**What happens**

When you click Yes, InfoMaker:

- ◆ Deletes the specified database
- ◆ Removes the data source from the ODBC preferences file
- ◆ Deletes the database profile from the InfoMaker initialization file

## Working with tables in the Database painter

When you open the Database painter, the Select Tables dialog box displays listing all the tables and views in the current database that you have access to (including tables and views that were not created using InfoMaker). You can create a new table or open existing tables.

### Opening a table

❖ **To open a database table:**

- 1 Click the Open button in the PainterBar.

*or*

Select Object>Select Tables from the menu bar.

The Select Tables dialog box displays.

- 2 (Optional) To display system tables, select the Show System Tables checkbox.

**FOR INFO** For information about system tables, see "About system tables" on page 82.

---

#### **InfoMaker caches the table list**

The first time InfoMaker has to build a list of tables in the database, it retrieves the list of tables from the database and displays it. InfoMaker saves this list internally and uses it the next time a list of tables needs to be displayed; this saves time, because InfoMaker doesn't have to regenerate what could be a very large list.

By default, the list is refreshed every 30 minutes (1800 seconds). You can specify a different refresh rate by setting the Refresh Table List property in the Database Preferences property sheet.

**FOR INFO** For information about changing Database preferences, see "Modifying database preferences" on page 77.

---

- 3 Select the tables to open by doing one of the following:
  - ◆ Click the name of each table you want to open in the list displayed in the Select Tables dialog box, then click Open to open the selected tables.
  - ◆ Double-click the name of each table you want to open. Each table is opened immediately. Then click Cancel to close the Select Tables dialog box.

Representations of the selected tables display in the Database painter workspace.

## About system tables

By default, InfoMaker shows only user-created tables in the Select Tables dialog box. If you select Show System Tables in the dialog box, InfoMaker also shows system tables.

Two kinds of system tables exist in the database:

- ◆ System tables provided by your DBMS (for more information, see your DBMS documentation)
- ◆ InfoMaker system tables

### About InfoMaker system tables

InfoMaker stores extended attribute information you provide when you create or modify a table (such as the text to use for labels and headings for the columns, validation rules, display formats, and edit styles) in the Powersoft repository.

For example, in the Employee table, one column name is Emp\_lname. A label and a heading for the column is defined for InfoMaker to use in reports. The column label is defined as Last Name:. The column heading is defined as Last Name. The label and heading are stored in the PBCatCol table in the Powersoft repository.

---

### About the Powersoft repository

The Powersoft repository system tables are maintained by InfoMaker. The repository contains information about database tables and columns. Repository information extends database definitions. Only InfoMaker users can enter information into the repository.

---

The Powersoft repository has five system tables, each of which stores Powersoft-specific information about extended attributes:

<b>This system table</b>	<b>Stores this Powersoft-specific information</b>
PBCatCol	Column data such as name, header and label for reports and forms, and header and label positions
PBCatEdt	Edit style names and definitions
PBCatFmt	Display format names and definitions
PBCatTbl	Table data such as name, fonts, and comments

<b>This system table</b>	<b>Stores this Powersoft-specific information</b>
PBCatVld	Validation rule names and definitions

You can open system tables in the Database painter just like other tables.

**FOR INFO** For more about the tables in the Powersoft repository, see the Appendix, "The Powersoft Repository".

## Modifying properties in the Database painter

You can modify the properties of any table in the workspace, and of any column in a table. Table properties include the fonts used for headers, labels, and data, and a comment that you can associate with the table. Column properties include the text used for headers and labels, display formats, validation rules, and edit styles used for data, and a comment you can associate with the column.

### Modifying table properties in the Database painter

❖ **To modify the properties of the table selected in the Database painter:**

- 1 In the Database painter, click the Table Properties button in the PainterBar.

The Table property sheet displays.

- 2 Select a tab and specify properties.

---

**The Table property sheet is DBMS-dependent**

If you are connected to a text-based ODBC data source, only the General and Font property pages display. The Primary Key property page does not display.

---

<b>Select this tab</b>	<b>To modify this property</b>
General	Comments associated with the table
Data Font	Font for data displayed in the Data Manipulation painter
Heading Font	Font for headers displayed in the Data manipulation painter
Label Font	Font for labels displayed in the Freeform format of the Data Manipulation painter

Select this tab	To modify this property
Primary Key	The table's primary key

FOR INFO For information about modifying primary keys, see "Defining primary keys" on page 104.

---

### Setting table properties in the Database or Table painter

The properties that you can set on the Table property sheet in the Database painter are the same as those you can set in the Table painter. However, the Table Properties property sheet in the Table painter has two additional tab pages, for foreign keys and indexes.

FOR INFO For more information, see "Modifying table properties in the Table painter" on page 94.

---

## Modifying column properties in the Database painter

### ❖ To modify the column properties of a table in the Database painter:

- 1 Display the column's popup menu and select Properties.  
The Column property sheet displays.
- 2 Select a tab and specify properties:

Select this tab	To modify this property
General	Comments associated with the column
Headers	The header text used in the Data Manipulation painter, the label text used in the freeform format of the Data Manipulation painter, and their positions
Display	Display format for the column and its justification, display height and width, and for character columns, the case and whether the column is a picture column
Validation	Validation rule for the column
Edit Style	Edit style for the column



### Setting column properties in the Database or Table painter

Most of the properties that you can set on the property sheet for a column correspond to the extended attributes you can set when creating or altering a table in the Table painter. However, there are some properties you can set in the Column property sheet that you cannot set in the Table painter.

For example, in the Table painter you have to select an existing display format, edit style, and validation rule, but in the Database painter you can specify new ones.

**FOR INFO** For more information about the column properties you can set in the Table painter, see "Specifying extended attributes" on page 92. For information about working with display formats, edit styles, and validation rules, see Chapter 6, "Displaying and Validating Data".

## Specifying additional properties for character columns

You can also set two additional properties for character columns on the Display property page: Case and Picture.

Specifying the displayed case

You can specify whether InfoMaker converts the case of characters for a column in a report or form.

### ❖ To specify how character data should be displayed:

- ◆ On the Display property page, select a value in the Case dropdown listbox:

Value	Meaning
Any	Characters are displayed as they are entered
UPPER	Characters are converted to uppercase
lower	Characters are converted to lowercase

Specifying a column as a picture

You can specify that a character column can contain names of picture files (BMP or, on supported platforms, WMF or PICT files).

❖ **To specify that column values are names of picture files:**

- 1 On the Display property page, select the Picture checkbox.  
When the Picture checkbox is selected, InfoMaker expects to find bitmap (BMP), Windows metafile (WMF), or Macintosh PICT filenames in the column and displays the contents of the picture file—not the name of the file—in reports and forms.  
Because InfoMaker cannot determine the size of the image until execution time, it sets both display height and display width to 0 when you check the Picture checkbox.
- 2 Enter the size and optionally the justification for the picture.

## Closing a table

You can remove a table from the workspace by selecting Close from its popup menu. This action only removes the table from the Database painter workspace. It does not drop (remove) the table from the database.

## Dropping a table

You can drop a table in the Database painter workspace. Dropping removes the table from the database.

❖ **To drop a table:**

- 1 Click the Drop button.  
*or*  
Select Drop Table from the table's popup menu.  
*or*  
Select Object>Drop from the menu bar.  
InfoMaker prompts you to confirm the deletion.
- 2 Click Yes.

## Deleting orphaned table information

If you drop a table outside InfoMaker, information will remain in the repository about the table, including extended attributes for the columns.

❖ **To delete orphaned table information from the repository:**

- 1 Select Design>Synchronize PB Attributes from the menu bar.

InfoMaker prompts you to confirm your action.

- 2 Click Yes.

InfoMaker deletes information about the table's attributes from the repository.

If you try to delete orphaned table information and there is none, a message tells you that synchronization is not necessary.

## Working with tables in the Table painter

In the Table painter, you can create a new table definition or alter an existing table definition. You can also modify table properties and work with indexes and keys.

### About the Table painter

Although the Table painter is not an editor, it has some features that are similar to an editor. Instead of working with text, you work with table columns. For example, you can copy a column and paste it in the current table definition or in a different table definition.

Working with more than one table

Each time you open the Table painter, you can select one table definition to change or create one new table definition. If you want to work on more than one table at a time, open a new instance of the Table painter for each table.

Two views in the Table painter

As you work in the Table painter, most of the time you'll be working in the workspace creating or altering a table definition. But you can leave the workspace and enter SQL Syntax view to see pending changes (in SQL syntax) to the table definition as you work.

**FOR INFO** For more information, see "Working in SQL Syntax view" on page 99.

How the Table painter and the Database painter work together

The Table painter and the Database painter are independent painters, but when they're open at the same time, they work together.

For example, when you create and save a table definition in the Table painter, the Table painter notifies the Database painter that a new table exists and the new table displays in all instances of the Database painter. If you change a column property in the Table painter—say you add an index—you can see the change in the Database painter.

### Creating a new table from scratch in the Table painter

You can create a new table in InfoMaker in the current database. The current database is the database to which InfoMaker is connected.

❖ **To create a table in the current database:**

- 1 Click the Table button in the PowerBar.

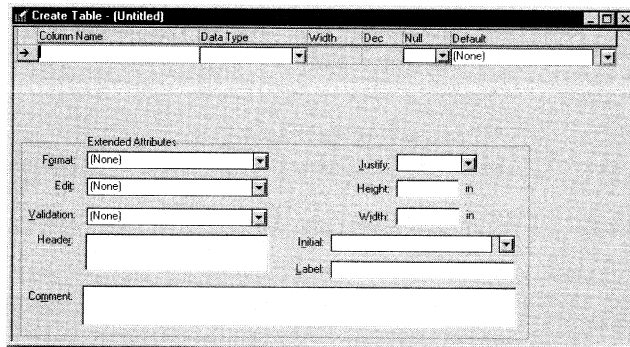
The Open Table dialog box displays.

**Displaying system tables**

You can open system tables (including the five tables in the Powersoft repository) in the Table painter. To see system tables in the Open Table list, select the Show System Tables checkbox. Don't change the definition of system tables. To guard against doing this, specify them as read-only by selecting the Read Only checkbox.

- 2 Click New.

The Table painter workspace displays with the insertion point in the Column Name box for the first column. The workspace is a template for specifying each column in the table. What you see in the workspace is DBMS-dependent:



- 3 Enter required information for the first column.

**FOR INFO** For what to enter in each field, see "Specifying the column definition" on page 91.

As you enter information, you can move from place to place in the column definition by pressing the TAB key. After defining the last item in the column definition (in this case, for SQL Anywhere, the Default definition), press the TAB key to display the work area for the next column.

- 4 (Optional) Specify extended attributes for the column.

You can do this now or later when you modify the table.

**FOR INFO** For what to enter in each field, see "Specifying extended attributes" on page 92.

- 5 Repeat steps 3 and 4 for each additional column in your table.
- 6 (Optional) Click the SQL Syntax button to see pending SQL syntax.  
The SQL syntax will be submitted to the database when you save or close the table. When you click the SQL Syntax button, you will be asked to enter a name for the table in the Create New Table dialog box.  
To return from SQL Syntax view to the Table painter workspace, click the SQL Syntax button again.
- 7 Click the Save button.  
If you did not do step 6, you will be asked to name the table. InfoMaker submits the pending SQL syntax statements it generated to the DBMS, and the table is created. Then control is returned to the Table painter with the new table open.
- 8 Click the Close button to close the table.  
If you have made no changes since you saved the table, the table closes. Otherwise, you are prompted to save the table.  
If you click Yes, InfoMaker applies all changes that are pending, and the Table painter closes.

---

#### **About saving the table**

If you make changes after you save the table and before you close it, you see the pending changes when you look at SQL Syntax view again. When you click Save again, InfoMaker submits a DROP TABLE statement to the DBMS, recreates the table, and applies all changes that are pending.

Clicking Save many times can be costly when you are working with large tables. So waiting to save a table until you have finished defining it can be a good practice.

---

## **Creating a new table from an existing table in the Table painter**

Because you can open a table as read-only, you can create a new table that's similar to an existing table very quickly by using the existing table in read-only mode as a template for the new table.

#### **❖ To create a new table using an existing table as a template:**

- 1 Click the Table button in the PowerBar.  
The Open Table dialog box displays.

- 2 Select the table you want to use as a template, select the Read Only checkbox, and then click Open.

The Table painter workspace displays the read-only table.

- 3 Make whatever changes you want to the table definition.
- 4 Click the Properties button to display the Table Properties property sheet.
- 5 On the General property page, change the table name and add comments to document what you've done.

**FOR INFO** For more information about modifying table properties, see "Modifying table properties in the Table painter" on page 94.

- 6 In the property sheet, click Apply to apply the changes you made to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.

*or*

Click OK to apply the request for changes and close the property sheet.

- 7 Click Save to save the table and then click Close to close the Table painter.

## Specifying column definitions and extended attributes

When you create a new table, you must specify a definition for the column. You can specify extended attributes for the column when you define it or you can do it later in the Table painter or the Database painter.

### Specifying the column definition

The fields that display for each column at the top of the Table painter depend on your DBMS. You may not see all of the following fields, and the values that you can enter are dependent on the DBMS.

**FOR INFO** For more information, see your DBMS documentation.

Field	What you enter
Column name	(Required) The name by which the column will be identified
Data Type	(Required) Select a data type from the dropdown listbox. All data types supported by the current DBMS are displayed in the listbox

Field	What you enter
Width	For data types with variable widths, the number of characters in the field
Dec	For numeric data types, the number of decimal places to display
Null	Select Yes or No from the Null dropdown listbox to specify whether NULLs are allowed in the column. Specifying No means the column cannot have NULL values; users must supply a value. No is the default in a new table
Default	The value that will be placed in a column in a row that you insert into a form. The dropdown listbox has built-in choices, but you can type any other value. For an explanation of the built-in choices, see your DBMS documentation

## Specifying extended attributes

In addition to providing the column information that is required to create a table, you can specify extended attributes for each column. An extended attribute is InfoMaker-specific information that enhances the definition of the column.

### ❖ To specify extended attributes in the Table painter workspace:

- 1 Display or create the table definition in the workspace.
- 2 Select the column to modify and specify the extended attributes.

The Extended Attributes group box at the bottom of the workspace is the area used for specifying the extended attributes. The following table shows the extended attributes that you can set in the Table painter:

Extended attribute	Meaning
Format	How the data is formatted in a report or form. For example, you can associate a display format with a Revenue column so that its data displays with a leading dollar sign, and negative numbers display in parentheses
Justify, Height, and Width	How data is aligned and how much space to allocate to the data in a report or form
Edit	How the column is presented in a report or form. For example, you can display column values as radio buttons or in a dropdown listbox



Extended attribute	Meaning
Validation	Criteria that a value must pass to be accepted in a form. For example, you can associate a validation rule with a Salary column so that you can only enter a value within a particular range
Initial	The initial value for the column. You can select a value from the list or type in a value. The initial value must be the same data type as the column, must pass validation, and can be NULL only if NULL is allowed for the column
Header	Header text used in tabular, grid, or n-up reports or in grid forms
Label	Label text used in freeform reports or forms
Comment	Description of the column. Whenever the table is opened in the Database painter, you can display column comments to understand the purpose of the columns

- 3 Click Save to save your changes.

---

### Specifying extended attributes in the Database painter

You can also specify extended attributes on the property sheet for the column in the Database painter.

**FOR INFO** For more information, see "Modifying column properties in the Database painter" on page 84.

---



---

### Overriding definitions

In the Report painter and Form painter, you can override the extended attributes specified in the Database or Table painter for a particular report or form.

---

How the information is stored

Extended attributes are stored in the Powersoft repository (in the InfoMaker system tables in the database). InfoMaker uses the information to display, present, and validate data in the Database painter, the Data Manipulation painter, and in reports and forms. When you create a view in the Database painter, the extended attributes of the table columns used in the view are used by default.

About display formats, edit styles, and validation rules

In the Table painter, you can select predefined display formats, edit styles, and validation rules from dropdown listboxes.

In the Database painter, you can create and name your own display formats, edit styles, and validation rules.

**FOR INFO** For more information about these extended attributes, see Chapter 6, "Displaying and Validating Data".

About headings and labels

By default, InfoMaker uses the column names as labels and headings, replacing any underscore characters with spaces and capping each word in the name. For example, the default label or heading for the column Dept\_name is Dept Name. To define multiple-line headings, press CTRL+ENTER to begin a new line.

---

#### **On Macintosh**

On the Macintosh, use COMMAND+RETURN to define multiple-line headings.

---

## **Modifying table properties in the Table painter**

### **❖ To modify the properties of the table selected in the Database painter:**

- 1 In the Table painter, click the Properties button in the PainterBar. The Table Properties property sheet displays.
- 2 Select a tab and specify properties.

---

#### **The Table property sheet is DBMS-dependent**

If you are connected to a text-based ODBC data source, only the General and Font property pages display. The Indexes and Key property pages do not display.

---

<b>Select this tab</b>	<b>To modify this property</b>
General	Comments associated with the table
Data Font	Font for data displayed in the Data Manipulation painter
Heading Font	Font for headers displayed in the Data manipulation painter
Label Font	Font for labels displayed in the freeform format of the Data Manipulation painter
Primary Key	The table's primary key

Select this tab	To modify this property
Foreign Keys	The table's foreign keys
Indexes	The table's indexes

3 Click Apply.

Any changes you've made in the property sheet are applied to the pending SQL Syntax changes that InfoMaker generates, not to the table definition.

You can view these pending changes by clicking the SQL Syntax button in the PainterBar. If you then click the Cancel button, the property sheet closes, but the specified pending changes already made are still pending.

4 Click Save.

The SQL executes. There are now no pending changes until you make additional changes.

---

### Specifying table properties in the Database painter

You can also specify table properties on the property sheet for the table in the Database painter.

**FOR INFO** For more information, see "Modifying table properties in the Database painter" on page 83.

---

## Altering a table in the Table painter

After a table is created, what you can do to alter the table depends on your DBMS.

In an existing table, you can always:

- ◆ Add or modify InfoMaker-specific extended attributes for columns
- ◆ Delete an index and create a new index

In an existing table, you can never:

- ◆ Insert a column between two existing columns
- ◆ Prohibit NULL values for an appended column
- ◆ Alter an existing index

In an existing table, some DBMSs let you do the following, but others do not:

- ◆ Append columns that allow NULLs
- ◆ Increase or decrease the number of characters allowed for data in an existing column
- ◆ Allow NULLs
- ◆ Prohibit NULLs in a column that allowed NULLs

---

**Table painter is DBMS-dependent**

The Table painter knows your DBMS and grays out actions or notifies you about actions that your DBMS prohibits.

---

**FOR INFO** If you need complete information about what you can and cannot do when you modify a table in your DBMS, see your DBMS documentation.

❖ **To alter a table:**

- 1 Open the Table painter and, in the Open Table dialog box, select the table you want to alter.

---

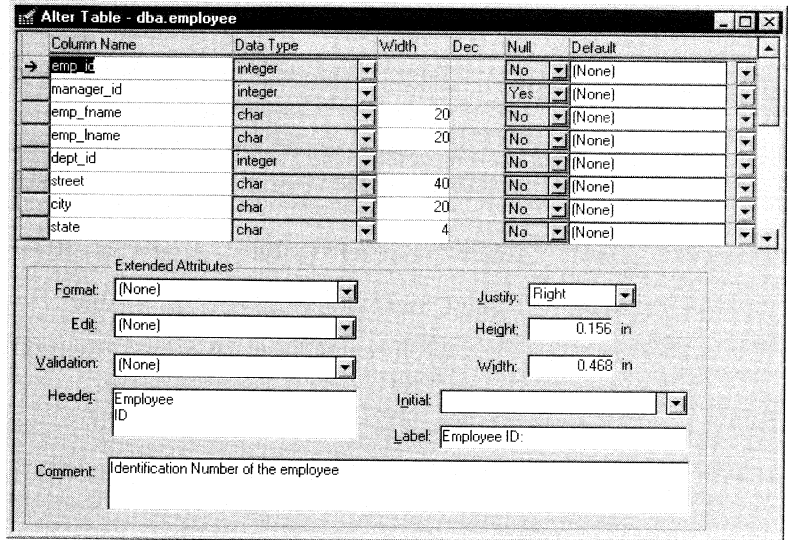
**Opening multiple instances of the Table painter**

In the Open Table dialog box, you can select only one table to open. However, you can click the Table button in the PowerBar at any time, select another table, and open another instance of the Table painter. Doing this is helpful when you want to use the Table painter's cut, copy, and paste features to cut or copy and paste between tables.

---

- Click Open.

The table definition displays in the Table painter workspace. This screen shows the Employee table:



- Make the changes you want in the workspace or in the Table Properties property sheet.

**FOR INFO** For information about settings you can change in the Table painter, see "Specifying extended attributes" on page 92. For information about the Table Properties property sheet, see "Modifying table properties in the Table painter" on page 94.

- Click the Save button.

InfoMaker submits the pending SQL syntax statements it generated to the DBMS, and the table is modified.

- Click the Close button.

## Cutting, copying, and pasting columns in the Table painter

In the Table painter, you can use the Cut, Copy, and Paste buttons in the PainterBar (or Edit>Cut Column, Edit>Copy Column, Edit>Paste Column in the menu bar) to cut, copy, and paste one column at a time within a table or between multiple tables.

❖ **To cut or copy a column within a table:**

- 1 Put the insertion point anywhere in the column you want to cut or copy.
- 2 Click the Cut or Copy button in the PainterBar.

❖ **To paste a column within a table:**

- 1 Put the insertion point in the column you want to paste to.  
If the table you want to paste a column to is an existing table, put the insertion point in the last column of the table. If you try to insert a column between two columns, you'll get an error message. You can only append a column to an existing table.
- 2 Click the Paste button in the PainterBar.

❖ **To paste a column to a different table:**

- 1 Open another instance of the Table painter and open a table or click New to create a new table.
- 2 Put the insertion point in the column you want to paste to.
- 3 Click the Paste button in the PainterBar.

## Specifying fonts for the table in the Table painter

When you create or alter a table, you can choose the fonts that will be used to display information from the table in a report or form.

❖ **To specify fonts for a table:**

- 1 In the Table painter workspace, click the Properties button.
- 2 On the Data Font, Heading Font, and Label Font property pages, specify the fonts as needed.

You can specify the font, point size, color, and style for the following:

- ◆ **Data** The values retrieved from the database
- ◆ **Headings** The column identifiers used in grid, tabular, and n-up reports and grid forms
- ◆ **Labels** The column identifiers used in freeform reports and forms

- 3 In the property sheet, click Apply to apply the changes you made to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.

*or*

Click OK to apply the request for changes and close the property sheet.

## Working in SQL Syntax view

Viewing pending SQL changes

As you create or alter a table definition in the Table painter, you can view the pending SQL syntax changes that will be made when you save the table definition.

### ❖ To view pending SQL syntax changes:

- ◆ Click the SQL Syntax button in the PainterBar.

*or*

Select Design>Syntax from the menu bar.

If you are working in a new table that hasn't been saved, InfoMaker asks you to name the table before displaying SQL syntax.

InfoMaker displays in SQL syntax the pending changes to the table definition:

```

ALTER TABLE "dba"."employee" DELETE "termination_date";

DELETE FROM "dba"."pbcatalcol"
  where pbc_tnam = 'employee'
 and pbc_owmr = 'dba'
 and pbc_cnam = 'termination_date';

CREATE UNIQUE INDEX "state"
  ON "dba"."employee"
  ("state");

```

The SQL statements execute only when you click the Save button to save the table definition or click the Close button and then tell InfoMaker to save changes.

Copying, saving, and printing pending SQL changes

When you are viewing pending SQL changes, you can:

- ◆ Copy pending changes to the clipboard
- ◆ Save pending changes to a file
- ◆ Print pending changes

---

**To copy, save, or print only part of the SQL syntax**

Select the part of the SQL syntax you want before you copy, save, or print.

---

❖ **To copy the SQL syntax to the clipboard:**

- ◆ In SQL Syntax view, click the Copy button.  
*or*  
Press CTRL+C.

---

**On Macintosh**

On the Macintosh, press the COMMAND key instead of the CTRL key.

---

The SQL syntax or the selected part is copied to the clipboard.

❖ **To save SQL syntax for execution at a later time:**

- 1 In SQL Syntax view, Select File>Save As.  
The Save Syntax to File dialog box displays.
- 2 Navigate to the folder in which you want to save SQL, name the file, and then click the Save button.  
The SQL syntax or the selected part is saved to the file.

At a later time, you can import the SQL file into the Database Administration painter and execute it.

❖ **To print pending table changes:**

- ◆ While viewing SQL syntax, select File>Print from the menu bar.  
The SQL syntax or the selected part is printed.

## Logging applied SQL syntax changes

You can choose to save to a log file the SQL syntax changes that have been applied to tables.

❖ **To save applied table changes to a log file:**

- 1 In the Table painter workspace, click the Properties button to display the Table Properties property sheet.
- 2 On the General property page, enter the name and location of the log file.



Now when you save changes to any table and close the Table painter, InfoMaker will ask if you want to save applied changes to the log file.

---

**Changes to other tables will be saved to the same log file**

All instances of the Table painter will continue to ask about saving applied table changes to the log file until you delete or change the log file specification in the Table Properties property sheet. If you don't delete or change the file specification, changes to many tables can be saved to the same log. A banner that displays the date and time precedes the applied changes made for each Table painter instance.

---

## Printing the table definition

You can print a report of the table's definition at any time, whether or not the table has been saved. The Table Definition Report contains information about the table and each column in the table, including the extended attributes for each column.

❖ **To print the table definition:**

- ◆ In the Table painter workspace, select File>Print from the menu bar.

## Working with keys

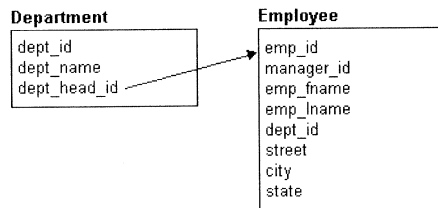
If your DBMS supports primary and foreign keys, you can work with the keys in InfoMaker. When you open a table with keys in the Database painter, InfoMaker gets the information from the DBMS and displays it in the painter workspace.

## Why you should use keys

If your DBMS supports them, you should use primary and foreign keys to enforce the referential integrity of your database. If you use keys, you can rely on the DBMS to make sure that only valid values are entered for certain columns.

For example, say you have two tables, Department and Employee. The Department table contains the column Dept\_Head\_ID, which holds the ID of the department's manager. You want to make sure that only valid employee IDs are entered in this column. That is, the only valid values for Dept\_Head\_ID in the Department table are values for Emp\_ID in the Employee table.

To enforce this kind of relationship, you define a foreign key for Dept\_Head\_ID that points to the Employee table. With this key in place, the DBMS disallows any value for Dept\_Head\_ID that does not match an Emp\_ID in the Employee table:



**FOR INFO** For more about primary and foreign keys, consult a book about relational database design or your DBMS documentation.

## What you can do in the painters

In this painter	You can
Database	Look at existing primary and foreign keys

In this painter	You can
Database	Open all tables that depend on a particular primary key
Database	Open the table containing the primary key used by a particular foreign key
Database and Table	Create, alter, and drop keys

For the most part, you work with keys the same way for each DBMS that supports keys. But there are some DBMS-specific issues.

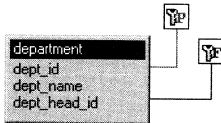
**FOR INFO** For complete information about using keys with your DBMS, see your DBMS documentation.

## Viewing keys

In the Database painter, when you open and expand a table containing primary and/or foreign keys, InfoMaker displays the keys in the workspace. The keys are shown as icons with lines connected to the table.

In the following picture, the Department table has two keys:

- ◆ A primary key (on dept\_id)
- ◆ A foreign key (on dept\_head\_id)



### If you can't see the lines

If the color of your window background makes it hard to see the lines for the keys and indexes, you can set the colors for each component of the Database painter's graphical table representation, including keys and indexes.

**FOR INFO** For information, see "Modifying database preferences" on page 77.

## Opening related tables

When working with tables containing keys, you can easily open related tables.

❖ **To open the table that a particular foreign key references:**

- 1 In the Database painter, open and expand the table containing the foreign key.
- 2 Display the popup menu for the button representing the foreign key.
- 3 Select Open Referenced Table from the popup menu.

❖ **To open all tables referencing a particular primary key:**

- 1 In the Database painter, open and expand the table containing the primary key.
- 2 Display the popup menu for the button representing the primary key.
- 3 Select Open Dependent Table(s) from the popup menu.

InfoMaker opens and expands all tables in the database containing foreign keys that reference the selected primary key.

## Defining primary keys

If your DBMS supports primary keys, you can define them in InfoMaker.

❖ **To create or modify a primary key:**

- 1 In the Database painter, click the Table Properties button.  
*or*  
In the Table painter, click the Properties button.
- 2 Select the Primary Key tab.
- 3 In the Table Columns box, select one or more columns for the primary key (or select and deselect as needed to modify an existing primary key).

Each time you select a key column, the column displays in the Key Columns box. Each time you deselect a key column, the column is removed from the Key Columns box.

---

**Columns that are allowed in a primary key**

Only a column that does not allow NULLs can be included as a column in a primary key definition. If you choose a column that allows NULLs, you will get a DBMS error when you save the table. In DBMSs that allow rollback for Data Definition Language (DDL), the table definition will be rolled back. In DBMSs that don't allow rollback for DDL, the painter is refreshed with the current definition of the table.

---

You can reorder the columns in the key by dragging them with the mouse.

- 4 Specify any information required by your DBMS.

---

**Naming a primary key**

Some DBMSs allow you to name a primary key and specify whether it is clustered or not clustered. For these DBMSs, the Primary Key property page has a way to specify these properties.

---

**FOR INFO** For DBMS-specific information, see your DBMS documentation.

- 5 In the property sheet, click Apply to apply the create- or drop-create-primary-key request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.  
*or*  
Click OK to apply the request for changes and close the property sheet.
- 6 Click Save.

---

**Completing the primary key**

Some DBMSs automatically create a unique index when you define a primary key so that you can immediately begin to add data to the table. Others require that you separately create a unique index to support the primary key before populating the table with data.

**FOR INFO** To find out what your DBMS does, see your DBMS documentation.

---

## Defining foreign keys

If your DBMS supports foreign keys, you can define them in InfoMaker.

❖ **To create a foreign key:**

- 1 In the Database painter, click the Create FK button.

*or*

In the Table painter, click the Properties button, select the Foreign Keys tab, and click New.

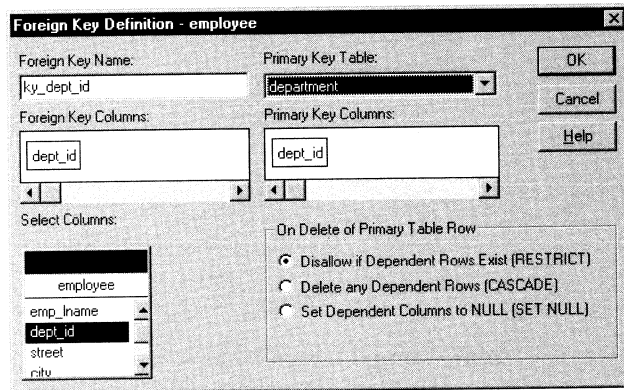
The Foreign Key Definition dialog box displays. Some of the information in the dialog box is DBMS-specific.

- 2 Name the foreign key in the Foreign Key Name box.
- 3 Select the columns for the foreign key in the Select Columns listbox.

The selected columns display in the Foreign Key Columns box. You can reorder the columns by dragging them with the mouse.

- 4 In the Primary Key Table listbox, select the table containing the Primary key referenced by the foreign key you are defining.

InfoMaker displays the selected table's primary key in the Primary Key Columns box:



---

**Key definitions must match exactly**

The definition of the foreign key columns must match the primary key columns, including datatype, precision (width), and scale (decimal specification).

---

- 5 Specify any information required by your DBMS.

For example, you may need to specify a delete rule by clicking a radio button in the On Delete of Primary Table Row groupbox.

**FOR INFO** For DBMS-specific information, see your DBMS documentation.

- 6 Click OK to close the dialog box.  
If you are working in the Database painter, you return to the workspace.  
If you are working in the Table painter, you return to the Table Properties property sheet.
- 7 Click Apply to apply the create-foreign-key request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.  
*or*  
Click OK to apply the request for changes and close the property sheet.
- 8 Click Save to save your changes.

## Dropping a key

You can drop keys (remove them from the database) from within InfoMaker.

❖ **To drop a key in the Database painter:**

- 1 In the Database painter, open the table containing the key.
- 2 Select Drop Primary Key or Drop Foreign Key from the key's popup menu.
- 3 Click Yes to confirm the deletion.

❖ **To drop a primary key in the Table painter:**

- 1 In the Table painter workspace, click the Properties button and then select the Primary Key tab.
- 2 In the Table Columns box, deselect all the columns in the primary key.
- 3 In the property sheet, click Apply to apply the drop-primary-key request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.

*or*

Click OK to apply the request for changes and close the property sheet.

- 4 Click Save.

❖ **To drop a foreign key in the Table painter:**

- 1 In the Table painter workspace, click the Properties button and then select the Foreign Keys tab.
- 2 Select a key from the Foreign Keys box and then click Delete.

- 3 In the property sheet, click Apply to apply the drop-foreign-key request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.  
*or*  
Click OK to apply the request for changes and close the property sheet.
- 4 Click Save.



## Working with indexes

You can work with indexes in the Database painter or the Table painter. You can create as many single- or multi-valued indexes for a database table as you need, and you can drop indexes that are no longer needed.

---

### Update limitation

You can update a table in a form only if it has a unique index or primary key.

---

## Creating an index

---

### In SQL Anywhere databases

You shouldn't define an index on a column that is defined as a foreign key, because foreign keys are already optimized for quick reference.

---

#### ❖ To create an index:

- 1 In the Database painter, click the Index button.

*or*

In the Table painter, click the Properties button, select the Indexes tab, and click New.

The Create Index dialog box displays. Some of the information is DBMS-specific, so the dialog box may look different for different DBMSs.

- 2 Enter a name for the index in the Index Name box.
- 3 Select whether or not to allow duplicate values for the index.
- 4 Specify any other information required for your database (for example, in SQL Server specify whether the index is clustered, and in SQL Anywhere specify the order of the index).
- 5 Click the names of the columns that make up the index.  
The selected column names display in the Index Columns box.
- 6 Click OK to close the dialog box.

If you are working in the Database painter, you return to the workspace.

If you are working in the Table painter, you return to the Table Properties property sheet.

- 7 Click Apply to apply the create-index request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.  
*or*  
Click OK to apply the request for changes and close the property sheet.
- 8 Click Save to save your changes.

## Modifying an index

Modifying an index is similar to creating an index, but you can only modify an index in the Table painter.

In the Database painter, you can delete the index and create a new one. To see the definition of an index in the Database painter, double-click its icon.

### ❖ To modify an index:

- 1 In the Table painter, click the Properties button and then select the Indexes tab.
- 2 Select an index from the Indexes box and click Edit.
- 3 In the Create Index dialog box, select or deselect columns as needed.
- 4 In the property sheet, click Apply to apply the drop-create-index request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.  
*or*  
Click OK to apply the request for changes and close the property sheet.
- 5 Click Save.

## Dropping an index

Dropping an index removes it from the database.

### ❖ To drop an index from a table in the Database painter:

- 1 In the Database painter workspace, display the popup menu for the icon of the index you want to drop.
- 2 Select Drop Index.
- 3 Click Yes to confirm the drop.

❖ **To drop an index from a table in the Table painter:**

- 1 In the Table painter workspace, click the Properties button and then select the Indexes tab.
- 2 Select the name of the index you want to drop.
- 3 Click Delete.
- 4 In the property sheet, click Apply to apply the delete-index request to the list of pending SQL statements to be submitted to the DBMS and then work on another property page.  
*or*  
Click OK to apply the request for changes and close the property sheet.
- 5 Click Save.

## Working with views

A view gives you a different (and usually limited) perspective of the data in one or more tables. Although you see existing views listed in the Select Tables dialog box, a view does not physically exist in the database as a table does. Each time you select a view and use the view's data, InfoMaker executes a SQL SELECT statement to retrieve the data and creates the view.

**FOR INFO** For more information about using views, see your DBMS documentation.

You can define and manipulate views in InfoMaker. Typically you use views for the following reasons:

- ◆ To give names to frequently executed SELECT statements.
- ◆ To limit access to data in a table. For example, you can create a view of all the columns in the Employee table except Salary. Users of the view can see and update all information except the employee's salary.
- ◆ To combine information from multiple tables for easy access.

In InfoMaker, you can create single- or multiple-table views. You can also use a view when you define data to create a new view.

You open and manipulate existing views in the Database painter. You define views in the View painter, which is similar to the Select painter.

**FOR INFO** For more information about the Select painter, see "Using SQL Select" on page 166.

---

### Updating views

Some views are logically updatable; some are not. Some DBMSs don't allow any updating of views. For the rules your DBMS follows, see your DBMS documentation.

---

## Opening a view

### ❖ To open a view:

- 1 From the Database painter, click the Open button.

*or*

Select Object>Select Tables from the menu bar.

The Select Tables dialog box displays listing all tables and views defined in the database.

- 2 Select the view.
- 3 Click Open.

## Creating a view

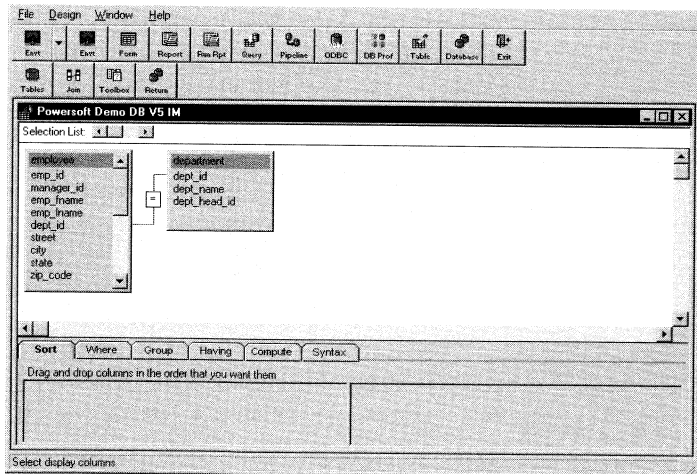
### ❖ To create a view:

- 1 From the Database painter, click the View button.  
*or*  
Select Object>New>View from the menu bar.

The Select Tables dialog box displays listing all tables and views that you can access in the database.

- 2 Select the tables and views from which you will create the view by doing one of the following:
  - ◆ Click the name of each table or view you want to open in the list displayed in the Select Tables dialog box, then click the Open button to open them. The Select Tables dialog box closes.
  - ◆ Double-click the name of each table or view you want to open. Each object is opened immediately. Then click the Cancel button to close the Select Tables dialog box.

Representations of the selected tables and views display in the View painter workspace:



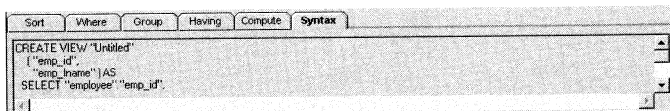
- 3 Select the columns to include in the view and include computed columns as needed.
- 4 Join the tables if there is more than one table in the view.  
FOR INFO For information, see "Joining tables" on page 115.
- 5 Specify criteria to limit (Where tab), group retrieved rows (Group tab), and limit the retrieved groups (Having tab) if appropriate.  
FOR INFO For information, see "Using SQL Select" on page 166. The View painter and the SQL Select painter are similar.
- 6 When the view has been completed, click the Return button.
- 7 Name the view.  
Include *view* or some other identifier in the view's name so that you will be able to distinguish it from a table in the Select Tables dialog box.
- 8 Click the Create button.  
InfoMaker generates a CREATE VIEW statement and submits it to the DBMS. The view definition is created in the database. You return to the Database painter workspace with the new view displayed in the workspace.

## Displaying a view's SQL statement

You can display the SQL statement that defines a view. How you do it depends on whether you are in the View painter creating a new view, or in the Database painter and want to look at the definition of an existing view.

❖ **To display the SQL statement from the View painter:**

- ◆ Select the Syntax tab in the View painter.



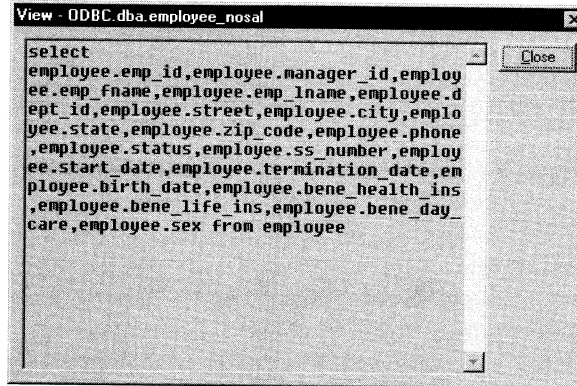
InfoMaker displays the SQL it is generating. The display is updated each time you change the view.

❖ **To display the SQL statement from the Database painter:**

- 1 Open the view.

- 2 Double-click the name of the view in the painter workspace.  
*or*  
Open the popup menu and select Definition.

InfoMaker displays the View dialog box showing the completed SELECT statement used to create the view:




---

#### View dialog box is read-only

You cannot alter the view definition in this dialog box. To alter a view, drop it and then create another view.

---

## Joining tables

If the view contains more than one table, you should join the tables on their common columns. When the View painter is first opened for a view containing more than one table, InfoMaker makes its best guess as to the join columns, as follows:

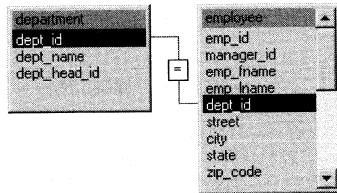
- ◆ If there is a primary/foreign key relationship between the tables, InfoMaker automatically joins them.
- ◆ If there are no keys, InfoMaker tries to join tables based on common column names and types.

#### ❖ To join tables:

- 1 Click the Join button.

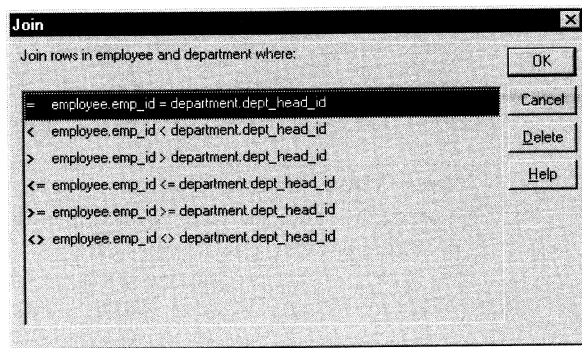
- 2 Click the columns on which you want to join the tables.

In the following screen, the Employee and Department tables are joined on the dept\_id column:



- 3 To create a join other than the equality join, click the join representation in the workspace.

The Join dialog box displays:



- 4 Select the join operator you want from the Join dialog box.

If your DBMS supports outer joins, outer join options also display in the Join dialog box (for example, in the preceding dialog box, which uses the Employee and Department tables, you can choose to include rows from the Employee table where there are no matching departments, or rows from the Department table where there are no matching employees).

---

### About the Query Governor

You can use the Query Governor to set data selection and retrieval options. If the Allow Cross Product option is set in the Query Governor, the Join dialog box displays outer join options.

FOR INFO For more information, see "Query Governor" on page 61.

---

FOR INFO For more about outer joins, see your DBMS documentation.



## Dropping a view

Dropping a view removes its definition from the database.

❖ **To drop a view:**

- 1 Select the view you want to drop in the Database painter workspace.
- 2 Click the Drop View button in the PainterBar.

InfoMaker prompts you to confirm the drop, then generates a DROP VIEW statement and submits it to the DBMS.

## Manipulating data

As you work on the database, you will often want to look at existing data or create some data for testing purposes. You will also want to test display formats, validation rules, and edit styles on real data.

InfoMaker provides the Data Manipulation painter for such purposes. In this painter, you can:

- ◆ Retrieve and manipulate database information
- ◆ Save the contents of the database in a variety of formats (such as Excel, dBASE, or Lotus 1-2-3)

### Opening the Data Manipulation painter

❖ **To open the Data Manipulation painter:**

- 1 In the Database painter, select the table or view whose data you want to manipulate.
- 2 Click the Grid, Table, or Freeform button in the PainterBar.

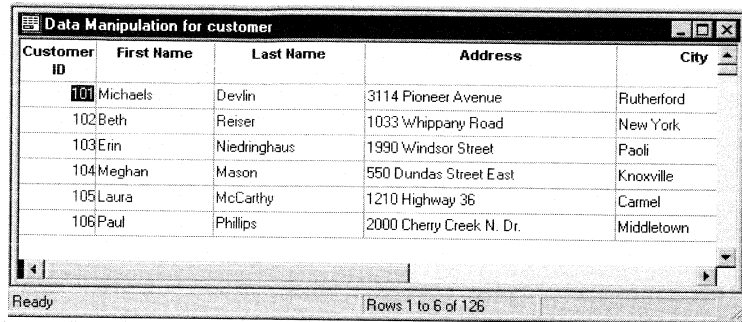
*or*

Select Object>Edit Data from the menu bar and choose one of the edit options from the cascading menu that displays.

The Data Manipulation painter opens and all rows are retrieved. As the rows are being retrieved, the Retrieve button changes to a Cancel button. You can click the Cancel button to stop the retrieval.

Exactly what you see depends on the formatting style you picked. The Data Manipulation painter is like a form. The formatting style you picked corresponds to a type of form (grid, tabular, or freeform). In a grid display, you can drag the mouse on a column's border to resize the column.

The window shown below is in the grid format:



Customer ID	First Name	Last Name	Address	City
101	Michaels	Devlin	3114 Pioneer Avenue	Rutherford
102	Beth	Reiser	1033 Whippary Road	New York
103	Erin	Niedringhaus	1990 Windsor Street	Paoli
104	Meghan	Mason	550 Dundas Street East	Knoxville
105	Laura	McCarthy	1210 Highway 36	Carmel
106	Paul	Phillips	2000 Cherry Creek N. Dr.	Middletown

Only a few rows of data display at a time. You can use the First, Prior, Next, and Last buttons to move from page to page.

## Retrieving data

### ❖ To retrieve rows from the database:

- ◆ Click the Retrieve button in the PainterBar.  
*or*  
Select Rows>Retrieve from the menu bar.

InfoMaker retrieves all the rows in the current table or view. As the rows are being retrieved, the button changes to Cancel. You can click it to stop the retrieval.

## Modifying data

You can add, modify, or delete rows. When you have finished manipulating the data, you can apply the changes to the database.

### If looking at data from a view

Some views are logically updatable; some are not. Some DBMSs don't allow any updating of views.

**FOR INFO** For the rules your DBMS follows regarding updating of views, see your DBMS documentation.

❖ **To modify data:**

- 1 Take one of the following actions:

<b>To do this</b>	<b>Take this action</b>
Modify existing data	Tab to a field and enter a new value
Add a row	Click the Insert Row button and enter data in the new row
Delete a row	Click the Delete Row button

The Data Manipulation painter uses validation rules, display formats, and edit styles that you or others have defined for the table in the Database painter when you add or modify data.

- 2 Click the Update Database button to apply changes to the database.

## Sorting and filtering data

You can define and use sort criteria and filters for the rows.

The sort criteria and filters you define in the Data Manipulation painter are for testing only and are not saved with the table or passed to the Report painter.

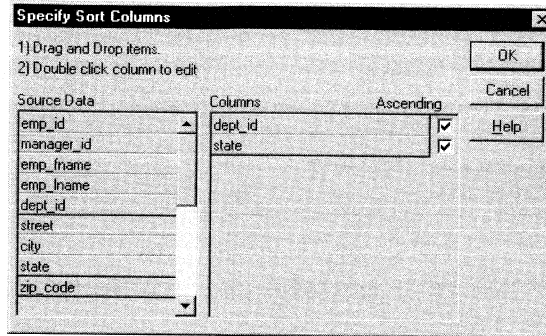
### Sorting rows

❖ **To sort the rows:**

- 1 Select Rows>Sort from the menu bar.

The Specify Sort Columns dialog box displays.

- 2 Drag the columns you want to sort on from the Source Data box to the Columns box:



A checkbox with a check in it displays under the Ascending heading to indicate that the values will be sorted in ascending order. To sort in descending order, clear the checkbox.

---

### Precedence of sorting

The order in which the columns display in the Columns box determines the precedence of the sorting. For example, in the preceding dialog box, rows will be sorted by department ID. Within department ID, rows will be sorted by state.

To change the precedence order, drag the column names in the Column box to the order you want.

- 3 (Optional) Double-click an item in the Columns box to specify an expression to sort on.

The Modify Expression dialog box displays.

- 4 Specify the expression.

For example, if you have two columns, Revenues and Expenses, you can sort on the expression *Revenues – Expenses*.

- 5 Click OK.

You return to the Specify Sort Columns dialog with the expression displayed.

---

### If you change your mind

You can remove a column or expression from the sorting specification by simply dragging it and releasing it outside the Columns box.

---

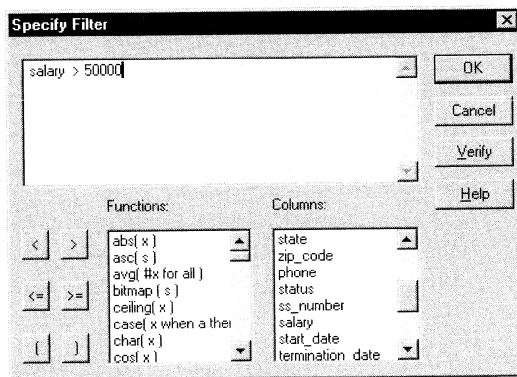
- 6 When you have specified all the sort columns and expressions, click OK.

## Filtering rows

You can limit which rows are displayed by defining a filter.

❖ **To filter the rows:**

- 1 Select Rows>Filter from the menu bar.  
The Specify Filter dialog box displays.
- 2 Enter a boolean expression that InfoMaker will test against each row:



If the expression evaluates to TRUE, the row will be displayed. You can paste functions, columns, and operators in the expression.

- 3 Click OK.  
InfoMaker filters the data. Only rows meeting the filter criteria are displayed.

❖ **To remove the filter:**

- 1 Select Rows>Filter from the menu bar.  
The Specify Filter dialog box displays showing the current filter.
- 2 Delete the filter expression, then click OK.

---

### Filtered rows and updates

Filtered rows are updated when you update the database.

---

## Viewing row information

You can display information about the data you have retrieved.

### ❖ To display row information:

- ◆ Select Rows>Described from the menu bar.

The Describe Rows dialog box displays showing the number of:

- ◆ Rows that have been deleted in the painter *but not yet deleted from the database*
- ◆ Rows displayed in Preview
- ◆ Rows that have been filtered
- ◆ Rows that have been modified in the painter *but not yet modified in the database*

All row counts are zero until you retrieve the data from the database or add a new row. The count changes when you modify the displayed data or test filter criteria.

## Importing data

You can import data from an external source and display it in the Data Manipulation painter, then save the imported data in the database.

### ❖ To import data:

- 1 Select Rows>Import from the menu bar.

The Select Import File dialog box displays.

- 2 Specify the file from which you want to import the data.

The types of files that you can import into the painter are shown in the Files of Type dropdown listbox.

- 3 Click Open.

InfoMaker reads the data from the file into the painter. You can then click the Update Database button in the PainterBar to add the new rows to the database.

## Printing data

You can print the data displayed in the Data Manipulation painter by selecting File>Print from the menu bar. Before printing, you can also preview the output on the screen.

❖ **To preview printed output before printing:**

- 1 Select File>Print Preview from the menu bar.  
Preview displays the data as it will print. To display rulers around the page borders in Print Preview, select File>Print Preview Rulers.
- 2 To change the magnification used in Print Preview, select File>Print Preview Zoom from the menu bar.  
The Zoom dialog box displays.
- 3 Select the magnification you want and click OK.  
Preview zooms in or out as appropriate.
- 4 When you have finished looking at the print layout, select File>Print Preview from the menu bar again.

## Saving data

You can save the displayed data in an external file.

❖ **To save the data in an external file:**

- 1 Select File>Save Rows As from the menu bar.  
The Save Rows As dialog box displays.
- 2 Choose a format for the file.  
You can select from several formats, including Powersoft report (PSR) or HTML.  
If you want the column headers saved in the file, select a file format that includes headers, such as Excel With Headers. When you select a *with headers* format, the names of the database columns (not the column labels) will also be saved in the file.  
**FOR INFO** For more information, see "Saving data in an external file" on page 212.



- 3 Name the file and save it.

InfoMaker saves all displayed rows in the file; all columns in the displayed rows are saved. Filtered rows are not saved.

## Returning to the Database painter workspace

- ❖ **To leave the Data Manipulation painter and return to the Database painter workspace:**

- 1 Select File>Close from the menu bar.

If you have made changes to the database but not yet saved them, InfoMaker asks you whether you want to update the database.

- 2 If you have unsaved changes, respond to the prompt.

You return to the Database painter.

## Administering the database

You can use the Database Administration painter to control access to the database and create SQL for immediate execution.

---

### Access to the Database Administration painter

You must install the Database painter to have access to the Database Administration painter. The Database painter is an optional InfoMaker painter.

---

## About administering databases

Administering your databases with the Database Administration painter means having the ability to:

- ◆ Create SQL statements to send to the DBMS for immediate execution
- ◆ Control access to the current database

### Building SQL statements

The Database Administration painter's workspace is a SQL editor in which you can enter SQL statements and execute them. The painter provides all editing capabilities needed for writing and modifying SQL statements. You can cut, copy, and paste text; search for and replace text; and paint SQL statements. You can also set editing properties to make reading your SQL files easier.

### Controlling database access

The Database Administration painter's Design menu provides access to a series of dialog boxes you can use to control access to the current database. For example, in some DBMSs you can assign table access privileges to users and groups.

Which menu items display on the Design menu and which dialog boxes display depend on your DBMS.

**FOR INFO** For information about support for security options in your DBMS, see *Connecting to Your Database* and your DBMS documentation.

## Opening the Database Administration painter

- ❖ **To open the Database Administration painter:**
  - ◆ Click the Database Administration button in the Database painter's PainterBar.  
*or*  
Select Design>Database Administration from the menu bar in the Database painter.

## Using the editor

The Database Administration painter provides the same editing capabilities as the file editor. It also has General, Font, and Coloring properties that you can change to make SQL files easier to read. With no change in properties, SQL files will have black text on a white background and a tab stop setting of 3 for indentation.

Select Design>Options from the menu bar to open the editor's property sheet. The General and Font properties are the same as those you can set for the File Editor.

**FOR INFO** For more information, see "Using the file editor" on page 33.

---

### Editor properties apply elsewhere

When you set General and Font properties for the Database Administration painter, the settings also apply to the file editor.

---

### Setting coloring properties

In addition to setting General and Font properties, you can set the text color and background color for SQL styles, such as data types and keywords, so that the style will stand out and the SQL code will be more readable. You set Coloring properties on the Coloring tab page of the Properties for Editor property sheet.

---

### Enabling syntax coloring

Be sure the Enable Syntax Coloring checkbox is selected before you set colors for SQL styles. You can turn off all Coloring properties by clearing the checkbox.

---

## Building and executing SQL statements

You can use the Database Administration painter to build SQL statements and execute them immediately. The painter's workspace acts as a notepad in which you can enter SQL statements.

Creating stored procedures

You can use the Database Administration painter to create stored procedures or triggers, but make sure that the painter's SQL statement terminator character is not the same as the terminator character used in the stored procedure language of your DBMS.

---

### About the statement terminator

By default, InfoMaker uses the semicolon as the SQL statement terminator. You can override the semicolon by specifying a different terminator character in the Database painter. To change the terminator character, select Design>Options from the Database painter's menu bar.

---

About comments

By default, InfoMaker strips off comments when it sends SQL to the DBMS. You can have comments included by clearing the check next to Design>Strip Comments from the menu bar.

Entering SQL

You can enter a SQL statement in three ways:

- ◆ Pasting the statement
- ◆ Typing the statement in the workspace
- ◆ Opening a text file containing the SQL

## Pasting SQL

You can paste SELECT, INSERT, UPDATE, and DELETE statements to the workspace. Depending on which kind of statement you want to paste, InfoMaker displays dialog boxes that guide you through painting the full statement.

### ❖ To paste a SQL statement to the workspace:

- 1 Click the Paste SQL button in the PainterBar.  
*or*  
Select Edit>Paste SQL from the menu bar.

The SQL Statement Type dialog box displays listing the types of SQL statements you can use.

- 2 Double-click the appropriate icon to select the statement type.

The Select Table dialog box displays.

- 3 Select the table(s) you will reference in the SQL statement.

You go to the Select, Insert, Update, or Delete painter, depending on the type of SQL statement you are pasting. The Insert, Update, and Delete painters are similar to the Select painter, but only the appropriate tabs display in the SQL toolbox at the bottom of the workspace.

**FOR INFO** For more information about the Select painter, see "Using SQL Select" on page 166.

- 4 Follow the procedure for the statement you are pasting:

Type of statement	What you do
SELECT	Define the statement exactly as in the Select painter when building a view. You choose the columns to select. If you want, you can define computed columns, specify sorting and joining criteria, and WHERE, GROUP BY, and HAVING criteria  <b>FOR INFO</b> For more information, see "Working with views" on page 112
INSERT	Type the values to insert into each column. You can insert as many rows as you want
UPDATE	First specify the new values for the columns in the Update Column Values dialog box. Then specify the WHERE criteria to indicate which rows to update
DELETE	Specify the WHERE criteria to indicate which rows to delete

- 5 When you have completed painting the SQL statement, click the Return button in the PainterBar in the Select, Insert, Update, or Delete painter.

You return to the Database Administration painter with the SQL statement pasted into the workspace.

## Typing SQL

If you want, you can simply type one or more SQL statements directly in the workspace.

You can enter most statements supported by your DBMS. This includes statements you can paint as well as statements you cannot paint (such as a database stored procedure or CREATE TRIGGER statement). You cannot enter certain statements that could destabilize the InfoMaker development environment. These include the SET statement and the USE *database* statement.

---

### **Sybase SQL server stored procedures**

When you use the Database Administration painter to execute a Sybase SQL Server system stored procedure, you *must* start the syntax with the keyword EXEC or EXECUTE. You cannot execute the stored procedure simply by entering its name. For example, enter:

```
EXEC SP_LOCK
```

---

## **Importing SQL from a text file**

You can import SQL that has been saved in a text file into the Database Administration painter.

### **❖ To read SQL from a file:**

- 1 Position the insertion point where you want to insert the SQL.
- 2 Select File>Import from the menu bar.  
The File Import dialog box displays.
- 3 Select the file containing the SQL and click OK.

## **Explaining SQL**

Sometimes there is more than one way to code SQL statements to obtain the results you want. When this is the case, you can use Explain SQL on the Design menu to help you select the most efficient method. Explain SQL displays information about the path that InfoMaker will use to execute the statements in the SQL Statement Execution Plan dialog box. This is most useful when you are retrieving or updating data in an indexed column or using multiple tables.

---

**DBMS-specific information**

The information displayed in the SQL Statement Execution Plan dialog box depends on your DBMS.

**FOR INFO** For more about the SQL execution plan, see your DBMS documentation.

---

## Executing SQL

When you have the SQL statements you want in the workspace, you can submit them to the DBMS.

❖ **To execute the SQL:**

- ◆ Click the Execute button.

*or*

Select Design>Execute SQL from the menu bar.

### What happens next

If the SQL retrieves data, the data appears in a window identical to a grid Data Manipulation painter.

**FOR INFO** For a description of what you can do with the data, see "Manipulating data" on page 118.

If there is a database error, you see a message box describing the problem.





PART 2

# Reports

This part introduces you to the many styles of reports available in InfoMaker and describes how to create and work with reports.



# Defining Reports

## About this chapter

The reports you create are centered around your organization's data. This chapter describes how to define reports to display the data.

## Contents

<b>Topic</b>	<b>Page</b>
Introducing reports	136
Building a report	143
Choosing a presentation style	145
Choosing report-wide options	153
Defining the data source	155
Generating and saving a report	185
Defining queries	188
What's next	191

## Introducing reports

Reports present data. The Report painter in InfoMaker provides many ways for you to present data. You may want a tabular report with rows and columns of information. Sometimes a graph or a crosstab is a better way to present the data.

An InfoMaker report can also be mailing labels or many reports nested together on the same page. Freeform InfoMaker reports let you place text, data, lines, boxes, and pictures anywhere you want. This means you can be very creative.

## Report examples

Following are sample InfoMaker reports that use data from the Powersoft Demo Database.

### Tabular report

The tabular report is the most common kind of report. You use it for presenting information in rows and columns. This is a basic tabular report:

List of Employees						
Employee Last Name	Employee First Name	Street	City	State	Zip Code	Phone
Evans	Scott	10-A Sunrise Circle	Concord	MA	01742-	(508) 555-0096
Kelly	Moir	12 Fountain Road	Gloucester	MA	01930-	(508) 555-3769
Letiecq	John	149 Vista Drive	Burlington	MA	01803-	(617) 555-1167
Lull	Kim	199 Lincoln Street	Concord	MA	01742-	(508) 555-4444
Pastor	Lynn	14 Cricklewood Drive	Burlington	MA	01803-	(617) 555-2001
Rabkin	Andrew	44 Birds Hill Way	Burlington	MA	01803-	(617) 555-4444
Savarino	Pamela	112 Beach Street	Long Beach	CA	90806-	(310) 555-1857
Scott	David	21 Riverdale Drive	Belmont	MA	02178-	(617) 555-3246
Shea	Mary Anne	197 Camden Road	Lexington	MA	02173-	(617) 555-4616
Sheffield	John	45 Belleview Drive	Houston	TX	77079-	(713) 555-3877
Shishov	Natasha	15 Milk Street	Waltham	MA	02154-	(617) 555-2755
Sterling	Paul	112 Endicott Street	Concord	MA	01742-	(508) 555-0295
Sullivan	Dorothy	124 Minuteman Drive	Lexington	MA	02173-	(617) 555-3947
Wang	Albert	48 Edwin Street	Waltham	MA	02154-	(617) 555-6741

A more advanced tabular report

This tabular report includes a column computed from other columns (Salary Plus Benefits) and special enhancements such as the *Confidential* watermark:

<b>Total Compensation Report</b>					Value of health ins. = \$4,800			Page 1 of 4
<b>Salary Plus Benefits</b>					Value of life insurance = \$(5.43 × salary)/1,000			3/22/97
					Value of day care = \$5,200			
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
100	102	Fran	Whitney	\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
	105	Matthew	Cobb	\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
	160	Robert	Breault	\$37,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,882
	243	Natasha	Shishoy	\$72,995	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
	247	Kurt	Driscoll	\$48,024	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
	249	Rodrigo	Guevara	\$42,998	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
	266	Ram	Gowda	\$59,840	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
	278	Terry	Melkisetian	\$48,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
	316	Lynn	Pastor	\$74,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905
	445	Kim	Lull	\$87,900	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$93,177
	453	Andrew	Rabkin	\$64,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$69,650

Grouped report

This Group style report groups by department and lists employees and salaries. It also includes a subtotal and a grand total for the salary column:

<b>Employee Report</b>				
4/8/97				
Department ID	Employee ID	First Name	Last Name	Salary
500				
	191	Jeannette	Bertrand	\$29,800
	703	Jose	Martinez	\$55,501
	750	Jane	Braun	\$34,300
	868	Felicia	Kuo	\$28,200
	921	Charles	Crowley	\$41,700
	1013	Joseph	Barker	\$27,290
	1570	Anthony	Rebeiro	\$34,576
	1615	Sheila	Romero	\$27,500
	1658	Michael	Lynch	\$24,903
<b>Total for department:</b>				<b>\$303,770</b>
<b>Grand Total:</b>				<b>\$3,749,147</b>

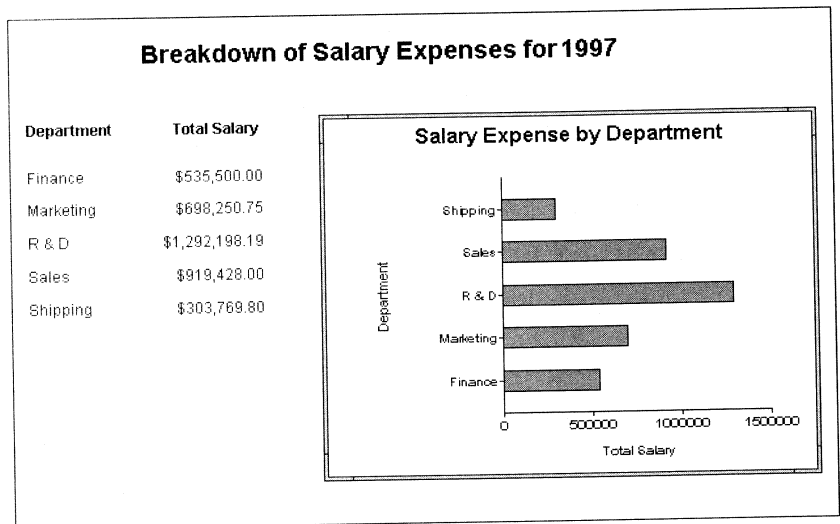
Grid report

This grid report looks very much like a tabular report. However, the grid is a rigid structure of rows and columns. You can change the column width and reorder the columns while you are viewing retrieved data. The grid report is useful for ad hoc reporting:

Employee Last Name	Employee First Name	Street	City	State	Zip Code	Phone
Ahmed	Alex	114 Cushing Street	Needham	MA	02192-	(617) 555-8748
Barker	Joseph	58 West Drive	Bedford	MA	01730-	(617) 555-8021
Barletta	Irene	37 Gleason Street	Bedford	MA	01730-	(617) 555-8345
Bertrand	Jeannette	209 Concord Street	Acton	MA	01720-	(508) 555-8138
Bigelow	Janet	84 Lunda Street	Waltham	MA	02154-	(617) 555-1493
Blaikie	Barbara	66 Beaumont Terrace	Needham	MA	02192-	(617) 555-9345
Braun	Jane	45 Wood Street	Cambridge	MA	02140-	(617) 555-7857
Breault	Robert	58 Cherry Street	Milton	MA	02186-	(617) 555-3099
Bucceri	Matthew	57 Taylor Place	Lexington	MA	02173-	(617) 555-5336
Butterfield	Joyce	119 Adams Street	Cambridge	MA	02140-	(617) 555-2232
Chao	Shih Lin	59 Holyoke Street	Lexington	MA	02173-	(617) 555-5921
Charlton	Doug	57 Webster Street	Concord	MA	01742-	(508) 555-9246
Chin	Philip	59 Pond Street	Atlanta	GA	30339-	(404) 555-2341
Clark	Alison	56 Carver Street	Emeryville	CA	94608-	(510) 555-9437

Report and graph

This report lists the salary totals by department. The graph presents the same data in a visual way, which makes it easier to see the relative cost of personnel in the five departments:



Freeform report













This freeform report presents all of the information about employees, one at a time. You can move the information around easily until you get what you want. The bitmap in the background marks the information as confidential:

### Employee Information

<p><b>Employee ID:</b> 102</p> <p><b>Manager ID:</b> 501</p> <p><b>Emp. First Name:</b> Fran</p> <p><b>Emp. Last Name:</b> Whitney</p> <p><b>Department ID:</b> 100</p> <p><b>Street:</b> 49 East Washington Street</p> <p><b>City:</b> Needham</p> <p><b>State:</b> MA</p> <p><b>Zip Code:</b> 02192-</p> <p><b>Phone:</b> (617) 555-3985</p>	<p><b>Status:</b> <input checked="" type="radio"/> Active  <input type="radio"/> Terminated  <input type="radio"/> On Leave</p> <p><b>Soc. Sec. No.:</b> 017-34-9033</p> <p><b>Salary:</b> \$45,700.00</p> <p><b>Start Date:</b> 8/28/1984</p> <p><b>Termination Date:</b> 0/00/0000</p> <p><b>Birth Date:</b> 6/05/1958</p> <p><b>Health Insurance:</b> <input checked="" type="checkbox"/></p> <p><b>Life Insurance:</b> <input checked="" type="checkbox"/></p> <p><b>Day Care:</b> <input type="checkbox"/></p>
--	---

Mailing labels

These mailing labels use the name and address information from the employee table and a bitmap to mark them with a logo:

 Alex Ahmed 114 Cushing Street Needham, MA 02192	 Joseph Barker 58 West Drive Bedford, MA 01730	 Irene B 37 Gle Bedford
 Jeannette Bertrand 209 Concord Street Acton, MA 01720	 Janet Bigelow 84 Lunda Street Waltham, MA 02154	 Barbar 66 Bea Needh
 Jane Braun 45 Wood Street Cambridge, MA 02140	 Robert Breault 58 Cherry Street Milton, MA 02186	 Matthe 57 Tay Lexing
 Joyce Butterfield 119 Adams Street Cambridge, MA 02140	 Shih Lin Chao 59 Holyoke Street Lexington, MA 02173	 Doug 57 We Conco

N-up report

This n-up report shows four rows of information next to each other. Similar to the freeform report, n-up is useful for fitting more information on the page. N-up is also useful for presenting periodic information, such as data that repeats for Monday through Friday (five blocks):

<b>Contacts</b>				
<i>Last</i> <i>First</i> <i>Phone:</i> <i>Fax:</i>	Bertrand Coleman (706) 555-2886 (704) 555-4532	Brier Michael (617) 555-2398 (617) 555-3337	Burrill Dana (617) 555-7956 (617) 555-2398	Caruso William (617) 555-2144 (617) 555-1656
<i>Last</i> <i>First</i> <i>Phone:</i> <i>Fax:</i>	Chin David (617) 555-3378 (617) 555-4453	Clarke Molly (617) 555-4325 (617) 555-7638	Cobb Paul (404) 555-2239 (404) 555-8111	Cohen Paul (617) 555-8883 (617) 555-4499
<i>Last</i> <i>First</i> <i>Phone:</i> <i>Fax:</i>	Collins MaryBeth (617) 555-1193 (617) 555-9586	Critch Susan (508) 555-4829 (508) 555-3025	Crossland Ellen (617) 555-0004 (617) 555-8005	Crowley Charles (617) 555-1344 (617) 555-9877
<i>Last</i> <i>First</i> <i>Phone:</i> <i>Fax:</i>	Davidson Joann (510) 555-7363 (510) 555-9278	DeMarco Michael (617) 555-4400 (617) 555-7876	Dewey Michael (617) 555-9877 (617) 555-2322	Elkins John (603) 555-1200 (603) 555-0078
<i>Last</i> <i>First</i> <i>Phone:</i> <i>Fax:</i>	Evans Carrie (404) 555-1169 (404) 555-8244	Fish Jeffrey (617) 555-3528 (617) 555-3563	Galvin Liz (617) 555-9312 (617) 555-9870	Glassmann Beth (617) 555-0273 (617) 555-3933

Crosstab

This crosstab report counts the number of employees that fit into each cell. For example, there are three employees in department 100 who make between \$30,000 and \$39,999:

Number of employees by department and salary 30,000 includes up to 39,999	Dept Id					Total number of employees making the salary
	Salary	100	200	300	400	
20000				2	5	7
30000	3	8	2	5	2	20
40000	6	5	2	5	1	19
50000	4	3	3	2	1	13
60000	4	1		2		7
70000	2	1	1			4
80000	2	1				3
90000	1					1
130000			1			1
<b>Total number of employees in the department</b>		22	19	9	16	9



Composite report with nested reports

This composite report consists of three nested tabular reports. One of the tabular reports includes a graph. Composite reports are a way to show different reports together on the same page:

3/23/97 **Quick Reference Information**

Products and Current Inventory				
Product ID	Product Name	Product Description	Unit Price	Number In Stock
300	Tee Shirt	Tank Top	\$0.00	28
301	Tee Shirt	V-neck	\$14.00	54
302	Tee Shirt	Crew neck	\$14.00	75
400	Baseball Cap	Common Cap	\$0.00	112
401	Baseball Cap	Wool cap	\$10.00	12
500	Visor	Cloth Visor	\$7.00	36
501	Visor	Plastic Visor	\$7.00	28
600	Sweatshirt	Hooded Sweatshirt	\$24.00	30
601	Sweatshirt	Zipper Sweatshirt	\$24.00	32
700	Shirts	Common Shirts	\$15.00	80

Sales Representatives and Total Number of Orders			
Sales Rep ID	Name	Phone	Number of Orders
120	PHILIP CHIN	(404) 555-2341	57
105	Marc Dill	(517) 555-2144	50
200	Rollin Drotbey	(510) 555-7255	114
467	James Klobescher	(713) 555-8627	56
667	Mani Garcia	(713) 555-3431	54
600	Kathleen Potasz	(517) 555-3020	52
856	Samuel Singer	(508) 555-3255	56
002	Melva Kelly	(508) 555-3760	47
040	Ranvula Savarino	(310) 555-1857	53
1142	Allison Clark	(510) 555-0437	57
1556	Catherine Pickett	(517) 555-3478	53

Product Sales Summary				
Product ID	Product Name	Product Description	Quantity Sold	Dollars
300	Tee Shirt	Tank Top	2364	\$21,276
301	Tee Shirt	V-neck	2388	\$33,432
302	Tee Shirt	Crew neck	2148	\$30,072
400	Baseball Cap	Common Cap	3278	\$30,502
401	Baseball Cap	Wool cap	2701	\$27,010
500	Visor	Cloth Visor	2652	\$18,564
501	Visor	Plastic Visor	2508	\$17,556
600	Sweatshirt	Hooded Sweatshirt	3060	\$73,440
601	Sweatshirt	Zipper Sweatshirt	2724	\$65,376
700	Shirts	Common Shirts	4536	\$68,040

Product	Dollars
Zipper Sweatshirt	65376
Wool cap	27010
V-neck	33432
Tank Top	21276
Plastic Visor	17556
Hooded Sweatshirt	73440
Crew neck	30072
Common Shirts	68040
Common Cap	30502
Cloth Visor	18564

Freeform report with a nested report

This freeform report lists all information about a customer and includes a related nested report that lists all the orders that belong to the customer. This is an example of a master/detail relationship—one customer has many orders:

3/23/97 **Customers and Orders**

**Customer Information**

Customer ID: 105  
 First Name: Laura  
 Last Name: McCarthy  
 Address: 1210 Highway 36  
 City: Carmel  
 State: IN  
 Zip Code: 46032  
 Phone Number: (317) 555-8437  
 Company Name: Amo & Sons

**Order History**

Sales Order ID	Order Date	Sales Rep ID	Line #	Product ID	Quantity	Date Shipped
2006	09/28/95	299	1	300	48	09/28/95
2344	03/30/95	195	1	501	36	03/31/95
2454	06/16/95	299	1	501	36	06/17/95
2568	09/21/95	856	1	600	36	09/22/95
			2	601	36	09/22/95

## Reports versus DataWindow objects

Reports are the same as the DataWindow objects that PowerBuilder users can create in the PowerBuilder DataWindow painter, except that they cannot be updated. When you create a report in the Report painter, you are actually creating a nonupdatable DataWindow object.

About the term  
DataWindow in  
examples

Some of the examples in this book were captured in the PowerBuilder environment. In these examples, the title bar of a report will include the word DataWindow instead of Report.

---

### **If you need update capabilities**

If you need update capabilities, use the InfoMaker Form painter to create a form. Forms are for updating, whereas reports are read-only.

---

## Building a report

You use the Report painter to work with reports.

## Connecting to a database

To use the Report painter, you must be connected to the database whose data you will be accessing.

When you open the Report painter, InfoMaker connects you to the DBMS and database you used last. If you need to connect to a different database, do so before working with a report.

**FOR INFO** For information about changing your database connection, see *Connecting to Your Database*.

## Modifying an existing report

❖ **To modify an existing report:**

- 1 Click the Report button in the PowerBar.

The Select Report dialog box displays listing the reports in the current library.

- 2 Select the existing report and click OK.

InfoMaker opens the Report painter and displays the report in the workspace.

**FOR INFO** To learn how you can modify an existing report, see Chapter 5, "Enhancing Reports".

## Creating a new report

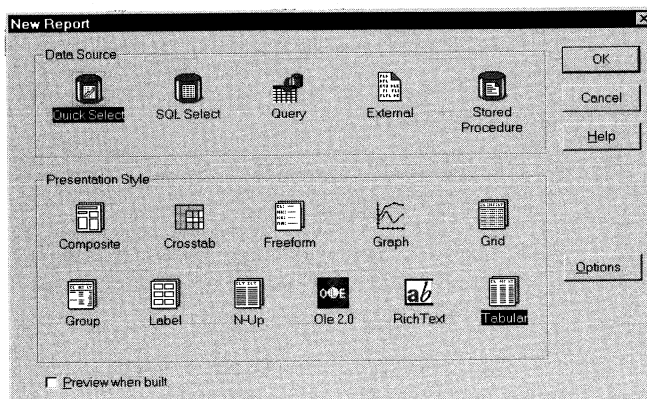
❖ **To create a new report:**

- 1 Click the Report button in the PowerBar.

The Select Report dialog box lists the reports in the current library.

- 2 Click the New button.

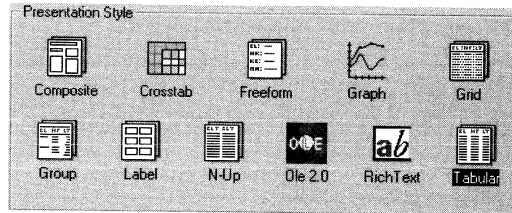
The New Report dialog box displays:



- 3 Choose a presentation style—how you want the data to be arranged—for the report.  
FOR INFO See "Choosing a presentation style" on page 145.
- 4 (Optional) Choose options for the report.  
FOR INFO See "Choosing report-wide options" on page 153.
- 5 Choose a data source for the report.  
FOR INFO See "Defining the data source" on page 155.
- 6 If you want to immediately preview (run) the basic report after defining the presentation style and data source, select the Preview When Built checkbox.  
After choosing and defining the data source, you go to the Report painter workspace. Although you can now save the report and begin using it, you may want to enhance it first.
- 7 (Optional) Enhance your report using the Report painter workspace.  
You can customize the display of column data and add objects such as text, bitmaps, computed fields, and graphs.  
You can also preview the report at any time in the Report painter.  
FOR INFO For more information about previewing, see Chapter 5, "Enhancing Reports".

## Choosing a presentation style

The presentation style you select for a report determines the format InfoMaker uses to first display the report in the Report painter workspace. You can use the format as displayed or modify it to meet your needs:

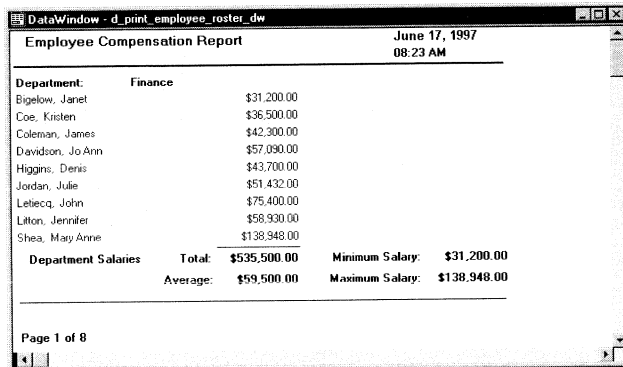


You can choose from the following presentation styles in the New Report dialog box:

- Tabular
- Freeform
- Grid
- Label
- N-Up
- Group
- Composite
- Graph
- Crosstab
- OLE 2.0 (on Windows)
- RichText

## Using the Tabular style

The Tabular presentation style presents data with the data columns going across the page and headers above each column. As many rows from the database will display at one time as can fit in the report. You can reorganize the default layout any way you want by moving columns and text. Tabular style is often used when you want to group data:

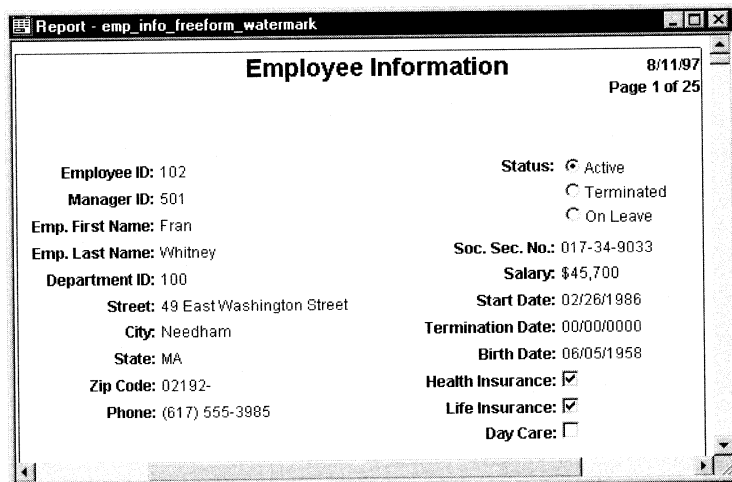


Employee Compensation Report		June 17, 1997 08:23 AM	
<b>Department:</b>	<b>Finance</b>		
Bigelow, Janet		\$31,200.00	
Coe, Kristen		\$36,500.00	
Coleman, James		\$42,300.00	
Davidson, Jo Ann		\$57,090.00	
Higgins, Denis		\$43,700.00	
Jordan, Julie		\$51,432.00	
Letiecq, John		\$75,400.00	
Litton, Jennifer		\$58,930.00	
Shea, Mary Anne		\$138,948.00	
<b>Department Salaries</b>	<b>Total:</b>	<b>\$535,500.00</b>	<b>Minimum Salary: \$31,200.00</b>
	<b>Average:</b>	<b>\$59,500.00</b>	<b>Maximum Salary: \$138,948.00</b>

Page 1 of 8

## Using the Freeform style

The Freeform presentation style presents data with the data columns going down the page and labels next to each column. You can reorganize the default layout any way you want by moving columns and text:



**Employee Information** 8/11/97  
Page 1 of 25

**Employee ID:** 102 **Status:**  Active  
 Terminated  
 On Leave

**Manager ID:** 501

**Emp. First Name:** Fran **Soc. Sec. No.:** 017-34-9033

**Emp. Last Name:** Whitney **Salary:** \$45,700

**Department ID:** 100 **Start Date:** 02/26/1986

**Street:** 49 East Washington Street **Termination Date:** 00/00/0000

**City:** Needham **Birth Date:** 06/05/1958

**State:** MA **Health Insurance:**

**Zip Code:** 02192- **Life Insurance:**

**Phone:** (617) 555-3985 **Day Care:**

## Using the Grid style

The Grid presentation style shows data in row-and-column format with grid lines separating rows and columns. With other styles, you can move text, values, and other objects around freely in designing the report. With the grid style, the grid lines create a rigid structure of cells.

An advantage of grid style is that you can reorder and resize columns when previewing the report.

Original Grid report

This grid report shows employee information. Several of the columns have a large amount of extra white space:

Employee ID	First Name	Last Name	Street	City	State
100	Fran	Whitney	49 East Washington Street	Needham	MA
109	Matthew	Cobb	77 Pleasant Street	Waltham	MA
129	Philip	Chan	69 Pond Street	Atlanta	GA
148	Julie	Jordan	144 Great Plain Avenue	Winchester	MA
160	Robert	Bresault	68 Cherry Street	Milton	MA
184	Melissa	Espinosa	112 Apple Tree Way	Stow	MA
191	Jeannette	Bertrand	209 Concord Street	Acton	MA
199	Marc	Dill	89 Hancock Street	Milton	MA
207	Jane	Francis	112 Hawthorne Drive	Concord	MA

Grid report with modified column widths













This grid report was created from the original one by decreasing the width of some columns:

Employee ID	First Name	Last Name	Street	City	State	Phone
100	Fran	Whitney	49 East Washington Street	Needham	MA	(617) 555-3985
109	Matthew	Cobb	77 Pleasant Street	Waltham	MA	(617) 555-3840
129	Philip	Chan	69 Pond Street	Atlanta	GA	(404) 555-2341
148	Julie	Jordan	144 Great Plain Avenue	Winchester	MA	(617) 555-7835
160	Robert	Bresault	68 Cherry Street	Milton	MA	(617) 555-3099
184	Melissa	Espinosa	112 Apple Tree Way	Stow	MA	(508) 555-2319
191	Jeannette	Bertrand	209 Concord Street	Acton	MA	(508) 555-8138
199	Marc	Dill	89 Hancock Street	Milton	MA	(617) 555-2144
207	Jane	Francis	112 Hawthorne Drive	Concord	MA	(508) 555-9022

## Using the Label style

The Label presentation style shows data as labels. With this style you can create mailing labels, business cards, name tags, index cards, diskette labels, file folder labels, and many other types of labels.

Mailing labels

 Rodrigo Guevara East Main Street Framingham, MA 01701	 Jeannette Bertrand 209 Concord Street Acton, MA 01720	 James Coleman 57 Heather Hill Drive Acton, MA 01720
 Joseph Barker 58 West Drive Bedford, MA 01730	 Irene Barletta 37 Gleason Street Bedford, MA 01730	 Robert Nielsen 55 Sargent Avenue Bedford, MA 01730
 Catherine Pickett 45 Appleton Road Bedford, MA 01730	 Sheila Romero 1 Oakview Terrace Bedford, MA 01730	 Doug Charlton 57 Webster Street Concord, MA 01742
 Scott Evans 10-A Sunrise Circle Concord, MA 01742	 Jane Francis 12 Hawthorne Drive Concord, MA 01742	 Jennifer Litton 17 Downing Street Concord, MA 01742

Business cards

<i>My company</i>  Terry Lambert Administration  204 Page St. Canton, MA 94608 Phone: (617) 555-2246 Fax: (617) 555-3692	<i>My company</i>  Terry Lambert Administration  204 Page St. Canton, MA 94608 Phone: (617) 555-2246 Fax: (617) 555-3692
<i>My company</i>  Terry Lambert Administration  204 Page St. Canton, MA 94608 Phone: (617) 555-2246 Fax: (617) 555-3692	<i>My company</i>  Terry Lambert Administration  204 Page St. Canton, MA 94608 Phone: (617) 555-2246 Fax: (617) 555-3692

Name tags

Lynn Page Sales	Charles Crowley Human resources
Dana Burrill Product development	William Caruso Finance

Specifying label properties

If you choose the Label style, you are asked to specify the properties for the label after specifying the data source. You can choose from a list of predefined label types or enter your own specifications manually.



Where label definitions come from

InfoMaker gets the information about the predefined label formats from a preferences file.

---

### Platform considerations

**On Windows** This preferences file is called PBLAB60.INI.

**On Macintosh** This preferences file is called InfoMaker Label Preferences.

---

## Using the N-Up style

The N-Up style presents two or more rows of data next to each other. It is similar to the Label style in that you can have information from several rows in the database across the page. However, the information is not meant to be printed on labels. The N-Up presentation style is useful if you have periodic data; you can set it up so each period repeats in a row.

After you select a data source, you are asked how many rows to display across the page.

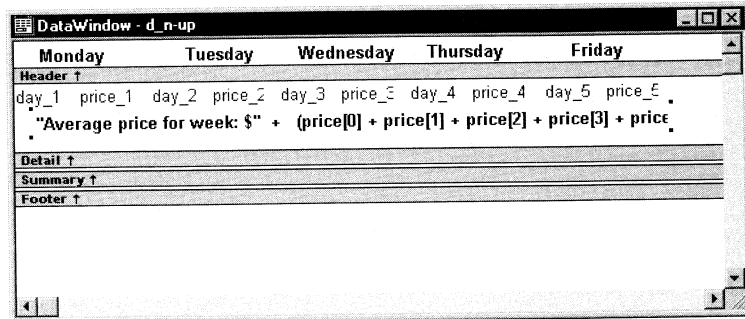
For each column in the data source, InfoMaker defines  $n$  columns in the report (column\_1 to column\_ $n$ ), where  $n$  is the number of rows you specified.

Table example

For a table of daily stock prices, you can define the report as five across, so each row in the report displays five days' prices (Monday through Friday). Here is a table with two columns that record the closing stock price each day for three weeks (this table is not in the Powersoft Demo Database):

Day	Price	Day	Price
1/31/96	62.00	2/10/96	68.00
2/1/96	63.00	2/11/96	67.00
2/2/96	66.00	2/14/96	67.00
2/3/96	65.00	2/15/96	71.00
2/4/96	62.00	2/16/96	70.00
2/7/96	65.00	2/17/96	72.00
2/8/96	69.00	2/18/96	75.00
2/9/96	66.00		

In the following n-up report, 5 was selected as the number of rows to display across the page, so each line in the report shows five days' stock prices. A computed field was added to get the average closing price in the week:



---

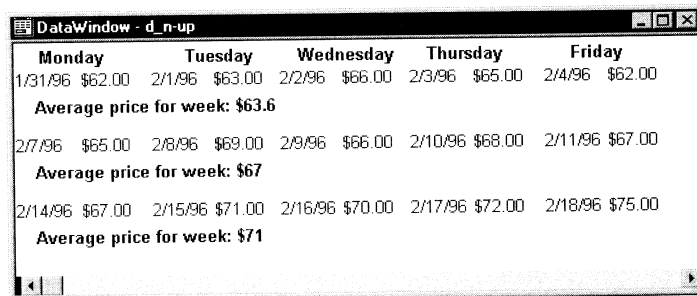
### About computed fields in n-up reports

You use subscripts, such as price[0], to refer to particular rows in the detail band in n-up reports.

FOR INFO For more information, see Chapter 5, "Enhancing Reports".

---

Here is the report in preview:



---

### Another way to get multiple-column reports

In an n-up report, the data is displayed across and then down. If you want your data to go down the page and then across in multiple columns, as in a phone list, you should create a standard tabular report, then specify newspaper columns.

FOR INFO For more information on newspaper columns, see Chapter 5, "Enhancing Reports".

---

## Using the Group presentation style

The Group presentation style provides an easy way to create grouped reports, where the rows are divided into groups, each of which can have statistics calculated for it. Using this style generates a tabular report that has grouping properties defined.

**FOR INFO** For more about the Group presentation style, see Chapter 7, "Filtering, Sorting, and Grouping Rows".

## Using the Composite presentation style

The Composite presentation style allows you to combine multiple reports in the same object. It is particularly handy if you want to print more than one report on a page.

**FOR INFO** For more about the Composite presentation style, see Chapter 9, "Using Nested Reports".

## Using the Graph and Crosstab presentation styles

In addition to the (preceding) text-based presentation styles, InfoMaker provides two styles that allow you to display information graphically: Graph and Crosstab.

**FOR INFO** For more information about these two presentation styles, see Chapter 10, "Working with Graphs", and Chapter 11, "Working with Crosstabs".

## Using the OLE 2.0 presentation style

The OLE presentation style lets you link or embed an OLE object in a report.

**FOR INFO** For information about the OLE 2.0 presentation style, see Chapter 13, "Using OLE in a Report".

## **Using the RichText presentation style**

The RichText presentation style lets you combine input fields that represent database columns with formatted text.

**FOR INFO** For more information about the RichText presentation style, see Chapter 12, "Working with Rich Text".

## Choosing report-wide options

You can preset the default options that InfoMaker uses in creating the initial draft of a report. These options include the default colors and borders that InfoMaker uses.

Report generation options are for styles that use the banded workspace, which include Freeform, Grid, Label, N-Up, Tabular, Group, and Crosstab. InfoMaker maintains a separate set of options for each of these styles.

When you are working on the draft of your report in the workspace, you can override the values set with report generation options.

### ❖ To specify default colors and borders for a new report:

- 1 In the New Report dialog box, click Options.

The Report Options property sheet displays.

---

#### In the painter

If you are already in the Report painter, you can select **Design>Options** from the menu bar to display the property sheet. Changes you make will apply to new reports. They will not apply to the open report.

---

- 2 Select the Generation tab page if it is not on top.
- 3 Select the presentation style you want from the Presentation Style dropdown listbox.

The values for properties shown on the page are for the currently selected presentation style.

- 4 Change one or more of the following properties:

Property	Meaning for the draft report
Background color	The default color for the background
Text border and color	The default border and color used for labels and headings
Column border and color	The default border and color used for data values

<b>Property</b>	<b>Meaning for the draft report</b>
Wrap Height (Freeform only)	The height of the detail band  When the value is None, the number of columns selected determines the height of the detail band. The columns display in a single vertical line  When the value is set to a number, the detail band height is set to the number specified and columns wrap within the detail band

5 Click OK.

You return to the New Report dialog box.

Your choices are saved

InfoMaker saves your generation option choices as the defaults to use when creating a report with the same presentation style.

## Defining the data source

The data source you choose determines how you select the data that will be used in the report.

### About the term *data source*

The term *data source* used here refers to how you use the Report painter to specify the data to retrieve into the report.

Data source can also refer to where the data comes from, such as a SQL Anywhere data source (meaning a database file) or a dBASE data source (meaning a DBF file). *Connecting to Your Database* uses the term data source this second way.

### InfoMaker data sources

InfoMaker has five data sources. All five can be used for reports, but only a subset of the possible data sources can be used for forms and data pipelines:

Data source	Reports	Forms	Pipelines
Quick Select	X	X	X
SQL Select	X	X	X
Query	X	X	X
External	X		
Stored Procedure	If the DBMS supports stored procedures that return result sets		X

### About stored procedures

The Stored Procedure data source icon displays only if the DBMS you are currently connected to supports stored procedures that return result sets.

You cannot create a form for data that is not stored in a database. A form allows you to display and change data in a database.

## How to choose the data source

To specify the data for a report, choose one of the following data sources:

<b>Data source</b>	<b>Use when</b>
Quick Select	The data is from a single table (or from tables that are related through foreign keys) and you only need to choose columns, selection criteria, and sorting (you don't need to specify grouping, computed columns, and so on)
SQL Select	You want more control over the SQL SELECT statement generated for the data source <i>or</i> your data is from tables that are not connected through a key
Query	The data has been defined as a query
External	The data is coming from text file (TXT) or dBASE II or dBASE III file (DBF)
Stored Procedure	The data is defined in a stored procedure

After you choose a data source in the New Report dialog box and click OK, you specify the data. The data source you choose determines which dialog box displays and how you define the data.

## Using Quick Select

The easiest way to define a data source is using Quick Select. With Quick Select, you can choose columns from one table or from multiple tables if they are joined through foreign keys. After you choose the columns, you can specify:

- ◆ Whether you want to sort the retrieved rows
- ◆ Retrieval criteria for the rows

### Quick Select limitations

When you choose Quick Select as your data source, you cannot:

- ◆ Specify grouping before rows are retrieved
- ◆ Include computed columns
- ◆ Specify retrieval arguments

To use these options, choose SQL Select as your data source.



❖ **To define the data source using Quick Select:**

- 1 Click Quick Select in the New Report dialog box, select a presentation style, and click OK.

The Quick Select dialog box displays. The Tables box lists tables and views in the current database.

---

**Which tables and views display**

The DBMS determines what tables and views display. For some DBMSs, all tables and views display, whether or not you have authorization. Then if you select a table or view you aren't authorized to access, the DBMS issues a message.

For ODBC databases, the tables and views that display depend on the driver for the data source. SQL Anywhere does not restrict the display; so all tables and views display, whether or not you have authorization.

---

To display a comment about a table, position the pointer on the table and click the right mouse button *or* select the table.

---

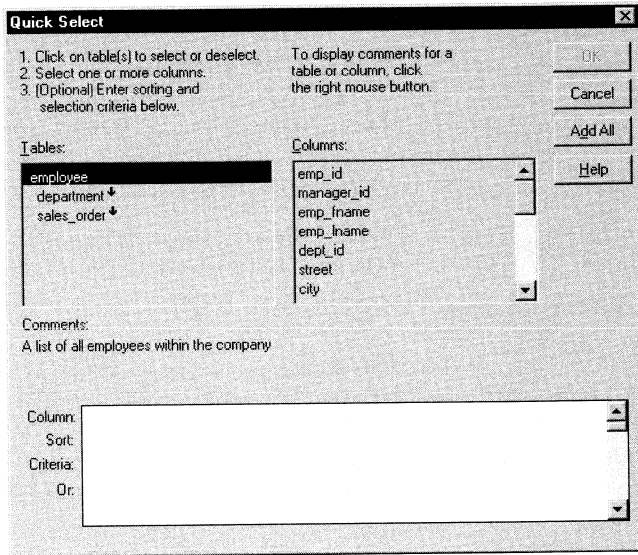
**On Macintosh**

To display a comment about a table on the Macintosh, position the pointer on the table, and press the CONTROL key while you click the mouse button *or* select the table.

---

- 2 Select a table containing the data you want to use.

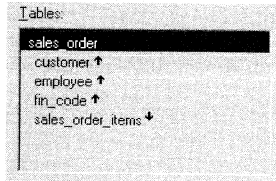
The table's column names display in the Columns box, and any tables having a key relationship with the selected table display in the Tables box. These tables are indented and marked with an arrow to show their relationship to the selected table:



---

### Meaning of the up and down arrows

An arrow displays next to a table to indicate its relationship to the selected table. The arrow always points in the *many* direction of the relationship—toward the selected table (up) if the selected table contains a foreign key in the relationship and away from the selected table (down) if the selected table contains a primary key in the relationship:



In this example, a foreign key in the sales\_order table is mapped to the primary key in the customer, employee, and fin\_code tables. The sales\_order\_items table contains a foreign key mapped to the primary key in the sales\_order table.

---

- 3 Select any additional tables containing data you want to use.

The column names of selected tables display in the Columns box. If you select more than one table, the column names are identified as:

*tablename.columnname*

For example, department.dept\_name and employee.emp\_id display when the Employee table and the Department table are selected.

---

### To return to the original table list

Click the table you first selected at the top of the table list.

---

- 4 Select the columns you want to use:
- ◆ To add a column, select it in the Columns box.
  - ◆ To add all columns, click Add All.
  - ◆ To remove a column, deselect it in the Columns box.
  - ◆ To view comments that describe a table or column, position the pointer on a table or column name, and press and hold the right mouse button.

---

**On Macintosh**

To view comments that describe a table or column on the Macintosh, position the pointer on the table or column name, hold down the CONTROL key, and press the mouse button.

---

The selected columns display at the bottom of the dialog box.

- 5 Use the grid to reorder columns, sort rows before you retrieve data, and specify what data to retrieve:

- ◆ To reorder a column, drag the column name where you want it.
- ◆ To specify the sorting of rows before retrieval, select the sorting order in the Sort row.

**FOR INFO** For information, see "Specifying sorting criteria" on page 160.

- ◆ To specify what data to retrieve, enter expressions in the Criteria row.

**FOR INFO** For information, see "Specifying selection criteria" on page 161.

- 6 Click OK.

The Quick Select dialog box closes.

In some report styles, InfoMaker prompts you for additional information.

The Report painter workspace displays.

---

**Quick Select and retrieval arguments**

When you use Quick Select to define the data, you cannot define retrieval arguments for the SELECT statement that are supplied during execution. If you decide later that you want to use retrieval arguments, you can define them from the Report painter workspace by modifying the data source.

**FOR INFO** For more information, see Chapter 5, "Enhancing Reports".

---

## Specifying sorting criteria

In the grid at the bottom of the Quick Select dialog box, you can specify if you want the retrieved rows to be sorted. As you specify sorting criteria, InfoMaker builds an ORDER BY clause for the SELECT statement.

❖ **To sort retrieved rows on a column:**

- 1 Click in the Sort row for the column you want to sort on.

InfoMaker displays a dropdown listbox:

Column:	Emp Id	Dept Id	Salary
Sort:		(not sorted) ▼	
Criteria:		Ascending	
Or:		Descending	
		(not sorted)	

- 2 Select the sorting order for the rows: Ascending or Descending.

**Multilevel sorts**

You can specify as many columns for sorting as you want. InfoMaker processes the sorting criteria left to right in the grid: the first column with Ascending or Descending specified becomes the highest level sorting column, the next column with Ascending or Descending specified becomes the next level sorting column, and so on.

If you want to do a multilevel sort that doesn't match the column order in the grid, drag the columns to the correct order and then specify the columns for sorting.

**Specifying selection criteria**

You can enter selection criteria in the grid to specify which rows to retrieve. For example, instead of retrieving data about all employees, you might want to limit the data to employees in Sales and Marketing or to employees in Sales and Marketing who make more than \$50,000.

As you specify selection criteria, InfoMaker builds a WHERE clause for the SELECT statement.

❖ **To specify selection criteria:**

- 1 Click the Criteria row below the first column for which you want to select the data to retrieve.
- 2 Enter an expression, or if the column has an edit style, select or enter a value.

If the column is too narrow for the criteria, drag the grid line to enlarge the column. This enlargement does not affect the column size in a report.

- 3 Enter additional expressions until you have specified the data you want to retrieve.

**About edit styles**

If a column has an edit style associated with it in the repository (that is, the association was made in the Database painter), the edit style is used in the grid—except that dropdown listboxes are used for columns with code tables and columns using the CheckBox and RadioButton edit styles.

SQL operators supported in Quick Select

You can use these SQL relational operators in the retrieval criteria:

Operator	Meaning
=	Is equal to (default operator)
>	Is greater than
<	Is less than
<>	Does not equal
>=	Is greater than or equal to
<=	Is less than or equal to
LIKE	Matches this pattern
NOT LIKE	Does not match this pattern
IN	Is in this set of values
NOT IN	Is not in this set of values

Because = is the default operator, you can enter the value *100* instead of = *100*, or the value *New Hampshire* instead of = *New Hampshire*.

Comparison operators

You can use the LIKE, NOT LIKE, IN, and NOT IN operators to compare expressions.

**The LIKE and NOT LIKE operators** Use LIKE to search for strings that match a predetermined pattern; use NOT LIKE to find strings that do not match a predetermined pattern. When you use LIKE or NOT LIKE, you can use wildcards:

- ◆ The percent sign (%), like the DOS wildcard asterisk (\*), matches multiple characters. For example, Good% matches all names that begin with Good.
- ◆ The underscore character (\_) matches a single character. For example, Good \_ \_ \_ matches all 7-letter names that begin with Good.

**The IN and NOT IN operators** Use IN to compare and include a value to a set of values; use NOT IN to compare and include values that are not in a set of values. For example, the following clause selects all employees in department 100, 200, or 500:

```
SELECT * from employee
WHERE dept_id IN (100, 200, 500)
```

Using NOT IN would exclude employees in those departments.

#### Connection operators

You can use the OR and AND logical operators to connect expressions.

InfoMaker makes some assumptions based on how you specify selection criteria. When you specify:

- ◆ **Criteria for more than one column on one line** InfoMaker assumes a logical AND between the criteria. A row from the database is retrieved if *all* criteria in the line are met.
- ◆ **Two or more lines of selection criteria** InfoMaker assumes a logical OR. A row from the database is retrieved if the criteria in *any* of the lines is met.

By default, criteria expressions in one line are logically ANDed; expressions in different lines are logically ORed. To override these defaults, begin an expression with the AND or OR operator:

Operator	Meaning
OR	The row is selected if one expression OR another expression is true
AND	The row is selected if one expression AND another expression are true

This technique is particularly handy when you want to retrieve a range of values in a column. See example 6 below.

#### SQL expression examples

**Example 1** The following expression in the grid retrieves information for employees whose salaries are less than \$50,000:

Column:	Emp Id	Dept Id	Salary
Sort:			
Criteria:			<50000
Or:			

The SELECT statement that InfoMaker creates is:

```
SELECT emp_id, dept_id, salary
FROM employee
WHERE salary < 50000
```

**Example 2** The following expression in the grid retrieves information for employees who belong to department 100:

Column:	Emp Id	Dept Id	Salary
Sort:			
Criteria:		100	
Or:			

The SELECT statement that InfoMaker creates is:

```
SELECT emp_id, dept_id, salary
FROM employee
WHERE dept_id = 100
```

**Example 3** The following expressions in the grid retrieve information for employees whose employee ID is greater than 300 *and* whose salary is less than \$50,000:

Column:	Emp Id	Dept Id	Salary
Sort:			
Criteria:	>300		<50000
Or:			

The SELECT statement that InfoMaker creates is:

```
SELECT emp_id, dept_id, salary
FROM employee
WHERE emp_id >300 AND salary <50000
```

**Example 4** The following expressions in the grid retrieve information for employees who either belong to:

- ◆ Department 100 *and* have a salary less than \$50,000

*or*

- ◆ A department whose ID is greater than 300, no matter what their salary

Column:	Emp Id	Dept Id	Salary
Sort:			
Criteria:		100	<50000
Or:		>300	

The SELECT statement that InfoMaker creates is:

```
SELECT emp_id, dept_id, salary
FROM employee
```



```
WHERE (dept_id = 100 AND salary < 50000)
OR dept_id > 300
```

**Example 5** The following expression in the grid retrieves information for employees who are in department 100 *or* 200 *or* 500:

Column:	Emp Id	Dept Id	Salary
Sort:			
Criteria:		IN (100,200,500)	
Or:			

The SELECT statement that InfoMaker creates is:

```
SELECT emp_id, dept_id, salary
FROM employee
WHERE dept_id IN (100, 200, 500)
```

**Example 6** The following expressions in the grid retrieve information for employees who have an employee ID from 500 to 1000 and a salary from \$30,000 to \$50,000:

Column:	Emp Id	Dept Id	Salary
Sort:			
Criteria:	>= 500		>= 30000
Or:	AND <= 1000		AND <= 50000

The SELECT statement that InfoMaker creates is:

```
SELECT emp_id, dept_id, salary
FROM employee
WHERE (emp_id >= 500 AND emp_id <= 1000)
AND (salary >= 30000 AND salary <= 50000)
```

**Example 7** The following expressions in the grid retrieve information for employees who have last names that begin with C or G:

Column:	Emp Last Name	Emp First Name	Salary
Sort:			
Criteria:	LIKE C%		
Or:	LIKE G%		

The SELECT statement that InfoMaker creates is:

```
SELECT emp_last_name, emp_first_name, salary
FROM employee
WHERE emp_last_name LIKE 'C%'
OR emp_last_name LIKE 'G%'
```

## Using SQL Select

In specifying your data for a report, you have the most freedom when you use SQL Select as the data source. When you choose SQL Select, you go to the Select painter, where you can paint a SELECT statement that includes the following:

- ◆ More than one table
- ◆ Selection criteria (WHERE clause)
- ◆ Sorting criteria (ORDER BY clause)
- ◆ Grouping criteria (GROUP BY and HAVING clauses)
- ◆ Computed columns
- ◆ One or more arguments to be supplied during execution

---

### **Saving your work as a query**

While in the Select painter, you can save the current SELECT statement as a query by selecting File>Save Query from the menu bar. Doing so allows you to easily use this data specification again in other reports.

**FOR INFO** For more information about queries, see "Defining queries" on page 188.

---

### **❖ To define the data using SQL Select:**

- 1 Click SQL Select in the New Report dialog box and click OK.  
The Select Tables dialog box displays.
- 2 Select the tables and/or views that you will use in the report.  
**FOR INFO** For more information, see "Selecting tables and views" on page 167.
- 3 Select the columns to be retrieved from the database.  
**FOR INFO** For more information, see "Selecting columns" on page 169.
- 4 Join the tables if you have selected more than one.  
**FOR INFO** For more information, see "Joining tables" on page 171.
- 5 Select retrieval arguments if appropriate.  
**FOR INFO** For more information, see "Using retrieval arguments" on page 173.

- 6 Limit the retrieved rows with WHERE, ORDER BY, GROUP BY, and HAVING criteria, if appropriate.

FOR INFO For more information, see "Specifying selection, sorting, and grouping criteria" on page 175.

- 7 If you want to eliminate duplicate rows, select Distinct from the Options menu. This adds the DISTINCT keyword to the SELECT statement.

- 8 Click the SQL Select button on the PainterBar.

The Report painter workspace displays (unless you need to provide more information, in which case you are prompted for it).

## Selecting tables and views

After you have chosen SQL Select in the New Report dialog box, chosen a presentation style, and clicked OK, the Select Tables dialog box displays in the current workspace.

What tables and views display depends on the DBMS. For some DBMSs, all tables and views display, whether or not you have authorization. Then if you select a table or view you aren't authorized to access, the DBMS issues a message.

For ODBC databases, the tables and views that display depend on the driver for the data source. SQL Anywhere does not restrict the display; so all tables and views display, whether or not you have authorization.

The SQL toolbox also displays at the bottom of the workspace; you will use the tabs in the SQL toolbox to specify the SQL Select statement in more detail.

### ❖ To select the tables and views:

- ◆ Do one of the following:
  - ◆ Click the name of each table or view you want to open.  
Each table you select is highlighted (to deselect a table, click it again). Click the Open button to close the Select Tables dialog box.
  - ◆ Double-click the name of each table or view you want to open.  
Each object opens immediately in the workspace behind the Select Tables dialog box. Click the Cancel button to close the Select Tables dialog box.

Representations of the selected tables and views display in the Select painter workspace. You can move or size each table to fit the workspace as needed.

Specifying what is displayed

You can display the label and data type of each column in the tables in the workspace (the label information comes from the Powersoft repository). If you need more space, you can choose to hide this information. You can also choose to hide the SQL toolbox to give you more room to see the tables.

❖ **To hide or display comments, data types, labels, and the SQL toolbox:**

- 1 Position the pointer on any unused area of the workspace and select Show from the popup menu.

A cascading menu displays.

- 2 Select or clear Datatypes, Labels, Comments, or SQL Toolbox as needed.

---

**Shortcut**

You can click the Toolbox button in the PainterBar to toggle the display of the SQL toolbox.

---

Colors in the Select painter

The colors used by the Select painter to display the workspace background and table information are specified in the Database painter. You can also set colors for the text and background components in the table header and detail areas.

**FOR INFO** For more information about specifying colors in the Database painter, see "Modifying database preferences" on page 77.

Adding and removing tables and views

At any time, you can add tables and views to your workspace:

To do this	Do this
Add tables or views	Click the Tables button in the PainterBar and select tables or views to add
Remove a table or view	Display its popup menu and select Close
Remove all tables and views	Select Design>Undo All from the menu bar

You can also remove individual tables and views from the workspace, or clear them all at once and begin selecting a new set of tables.

How InfoMaker joins tables

If you select more than one table in the Select painter workspace, InfoMaker joins columns based on their key relationship.

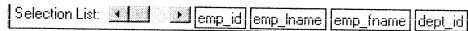
**FOR INFO** For information about joins, see "Joining tables" on page 171.

## Selecting columns

You can click each column you want to include from the table representations in the workspace. InfoMaker highlights selected columns and places them in the Selection List at the top of the Select painter.

### ❖ To reorder the selected columns:

- ◆ Drag a column in the Selection List with the mouse. Release the mouse button when the column is in the proper position in the list:

Selection List:  emp\_id | emp\_fname | emp\_fname | dept\_id

### ❖ To select all columns from a table:

- ◆ Move the pointer to the table name and select Select All from the popup menu.

### ❖ To include computed columns:

- 1 Click the Compute tab in the SQL toolbox at the bottom of the workspace.

Each row in the Compute tab is a place for entering an expression that defines a computed column.

- 2 Enter an expression for the computed column—for example:

```
salary / 12
```

or a function supported by your DBMS (the following is a SQL Anywhere function):

```
substr("employee"."emp_fname", 1, 2)
```

You can display the popup menu for any row in the Compute tab. Using the popup menu, you can select and paste columns, functions, and arguments (if you have created any) into the expression:

- ◆ Names of columns in the tables used in the report, form, or pipeline
- ◆ Any retrieval arguments you have specified
- ◆ Functions supported by the DBMS

---

**About these functions**

The functions listed here are functions provided *by your DBMS*. They are not InfoMaker functions. (This is because you are now defining a SELECT statement that will be sent to your DBMS for processing.)

---

- 3 Click the next row to define another computed column.

*or*

Click another tab to make additional specifications.

*or*

Click the SQL Select button to return to the workspace.

InfoMaker adds the computed columns to the list of columns you have selected.

---

**About defining computed columns here**

Computed columns you define in the Select painter are added to the SQL statement and used by the DBMS to retrieve the data. The expression you define here follows your DBMS's rules.

You can also choose to define computed fields, which are created and processed dynamically by InfoMaker after the data has been retrieved from the DBMS. There are advantages to doing this. For example, work is offloaded from the database server, and the computer fields update dynamically as data changes in the report (though if you have many rows, this updating can result in slower performance).

FOR INFO For more information, see Chapter 5, "Enhancing Reports".

---

## Displaying the underlying SQL statement

As you specify the data for the report in the Select painter, InfoMaker is generating a SQL SELECT statement. It is this SQL statement that will be sent to the DBMS when you retrieve data into the report. You can look at the SQL as it is being generated while you continue defining the data for the report.

❖ **To display the SQL statement:**

- ◆ Click the Syntax tab in the bottom of the workspace.

You may need to use the scroll bar to see all parts of the SQL SELECT statement. This statement is updated each time you make a change.

Editing the SELECT statement syntactically

Instead of modifying the data source graphically, you can directly edit the SELECT statement in the Select painter.

❖ **To edit the SELECT statement:**

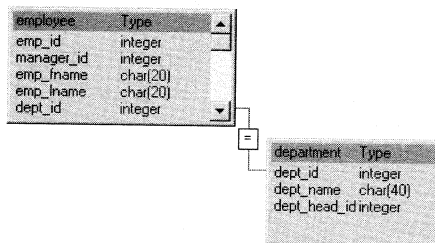
- 1 Select Design>Convert to Syntax from the menu bar.  
InfoMaker displays the SELECT statement in a text window.
- 2 Edit the SELECT statement.
- 3 Return to a painter:
  - ◆ **Select painter** Select Design>Convert to Graphics from the menu bar.
  - ◆ **Report painter** Click the SQL Select button.

## Joining tables

If the report will contain data from more than one table, you should join the tables on their common columns. If you have selected more than one table, InfoMaker joins columns based on their key relationship:

- ◆ **A primary/foreign key relationship** InfoMaker automatically joins them.
- ◆ **No key relationship** InfoMaker makes its best guess and tries to join tables based on common column names and types.

InfoMaker links joined tables in the Select painter workspace:



InfoMaker joins can differ depending on the order in which you select the tables, and sometimes the InfoMaker best-guess join is incorrect. So you may need to delete a join and manually define a join.

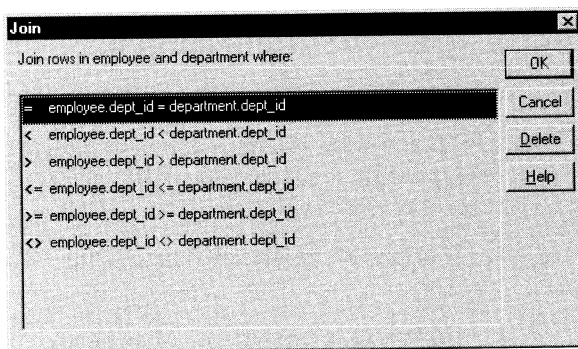
❖ **To delete a join:**

- 1 Click the join operator connecting the tables.  
The Join dialog box displays.
- 2 Click Delete.

❖ **To join tables:**

- 1 Click the Join button.
- 2 Click the columns on which you want to join the tables.
- 3 To create a join other than an equality join, click the join operator in the workspace.

The Join dialog box displays:



- 4 Select the join operator you want and click OK.

If your DBMS supports outer joins, and the Allow Cross Product option is set in the Query Governor, outer join options also display in the Join dialog box.

---

**About the Query Governor**

You can use the Query Governor to set data selection and retrieval options.

FOR INFO For more information about the Query Governor, see "Query Governor" on page 61.

---

FOR INFO For more about outer joins, see your DBMS documentation.



## Using retrieval arguments

If you want a report, form, or pipeline to prompt for criteria to determine which rows to retrieve when you preview the report, run the form, or execute the pipeline, you can use retrieval arguments in the SQL SELECT statement. If you define the data source without defining retrieval arguments and decide later that you need arguments, you can return to the Select painter from the painter workspace to define the arguments.

---

### Another way to prompt for retrieval criteria

You can select Rows>Prompt for Criteria from the menu bar. A dialog box lets you identify the column users should be prompted for. This, like the Retrieval Arguments prompt, calls the Retrieve function.

FOR INFO See Chapter 5, "Enhancing Reports" and Chapter 16, "Enhancing Forms".

---

For example, suppose you are creating a report that provides information about any employee. When you are defining the report in the Report painter, you pass the employee ID as an argument (placeholder). When you preview the report, you are prompted for the employee ID, you supply the ID number, and the report displays information about that employee.

#### ❖ To define retrieval arguments:

- 1 Make sure you are in the Select painter (from the Report painter or the Form painter, select Design>Edit Data Source from the menu bar).
- 2 In the Select painter, select Design>Retrieval Arguments from the menu bar.
- 3 Enter a name and data type for each argument.

The first character must be alphabetic (a-z); subsequent characters can be alphanumeric (a-z, 1-9), an underscore ( \_ ), or a dollar sign (\$).

- 4 Click the Add button to define additional arguments as needed and click OK when done.

#### Specifying an array as a retrieval argument

You can specify an array of values as your retrieval argument. For example, suppose you want a report that shows employee names and IDs for a few departments and prompts you to enter the IDs when you preview the report.

In the Specify Retrieval Arguments dialog box, choose the type of array from the Type dropdown listbox. For the case of department IDs, the array is a number array.

### Referencing retrieval arguments

After you define retrieval arguments, you must reference the arguments in the Where tab or Having tab in the Select painter.

To *reference an argument* means to refer to the argument in an expression so that InfoMaker can use it as a placeholder until you provide the actual value. For example, if a report is retrieving all rows from the Department table where the DeptID matches a value provided during preview, the WHERE clause looks something like this:

**WHERE DeptID = :Entered\_id**

where Entered\_id was defined previously as a retrieval argument in the Specify Retrieval Arguments dialog box.

---

#### How retrieval arguments are referenced

In SQL statements, variables (called host variables) are always prefaced with a colon to distinguish them from column names.

A retrieval argument is a variable. To reference the retrieval argument Entered\_id in a SQL statement, enter:

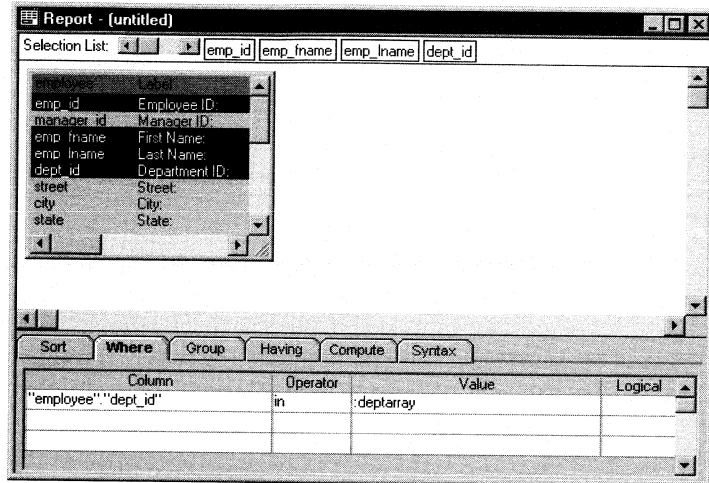
**:Entered\_id**

---

### Referencing an array

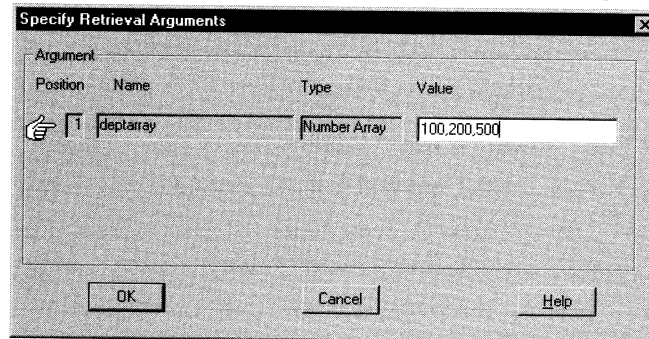
To reference an array, use the IN operator and reference the retrieval argument in the WHERE or HAVING clause.

For the case of the array defined as deptarray, the expression in the Where tab will look like the following expression. You can paste the :deptarray argument in using the popup menus in the value area of the Where tab:



Supplying values for an array argument

When you preview the report, you are prompted to supply the department values. InfoMaker retrieves rows that match one of the set of values that you supply. For example, if you supply the department IDs 100, 200, and 500 as shown, your report displays information about these departments:



## Specifying selection, sorting, and grouping criteria

In the SELECT statement associated with a report, you can add selection, sorting, and grouping criteria that are added to the SQL Statement and processed by the DBMS as part of the retrieval.

To do this	Use this clause
Limit the data that is retrieved from the database	WHERE
Sort the retrieved data before it is brought into the report	ORDER BY
Group the retrieved data before it is brought into the report	GROUP BY
Limit the groups specified in the GROUP BY clause	HAVING

---

### Dynamically selecting, sorting, and grouping data

Selection, sorting, and grouping criteria that you define in the Select painter are added to the SQL statement and processed by the DBMS as part of the retrieval. You can also define selection, sorting, and grouping criteria that are created and processed dynamically by InfoMaker *after* data has been retrieved from the DBMS.

**FOR INFO** For more information, see Chapter 7, "Filtering, Sorting, and Grouping Rows".

---

### Defining WHERE criteria

You can limit the rows that are retrieved into the report by specifying selection criteria that correspond to the WHERE clause in the SELECT statement.

For example, if you are retrieving information about employees, you can limit the employees to those in Sales and Marketing, or to those in Sales and Marketing who make more than \$50,000.

#### ❖ To define WHERE criteria:

- 1 Click the Where tab in the SQL Toolbox.

Each row in the Where tab is a place for entering an expression that limits the grouping of rows.

- 2 Click in the first row under Column to display columns in a dropdown list.

*or*

Select Columns from the popup menu.

- 3 Select the column you want to use in the left-hand side of the expression.

The equality (=) operator displays in the Operator column.

**Using a function or retrieval argument in the expression**

To use a function, select Functions from the popup menu and click a listed function. These are the functions provided by the DBMS.

To use a retrieval argument, select Arguments from the popup menu. You must have defined a retrieval argument already.

- 4 (Optional) Change the default equality operator.  
Enter the operator you want, or click to display a list of operators and select an operator.
- 5 Under Value, specify the right-hand side of the expression. You can:
  - ◆ Type a value.
  - ◆ Paste a column, function, or retrieval argument (if there is one) by selecting Columns, Functions, or Arguments from the popup menu.
  - ◆ Paste a value from the database by selecting Value from the popup menu, then selecting a value from the list of values retrieved from the database. (It may take some time to display values if the column has many values in the database.)
  - ◆ Define a nested SELECT statement by selecting Select from the popup menu. In the Nested Select dialog box you can define a nested SELECT statement. Click Return when you have finished.
- 6 Continue to define additional WHERE expressions as needed.  
For each additional expression, select a logical operator (AND or OR) to connect the multiple boolean expressions into one expression that InfoMaker evaluates as true or false to limit the rows that are retrieved.
- 7 Define sorting (Sort tab), grouping (Group tab), and limiting (Having tab) criteria as appropriate.
- 8 Click the SQL Select button to return to the Report painter workspace.

**Defining ORDER BY criteria**

You can sort the rows that are retrieved into the report by specifying columns that correspond to the ORDER BY clause in the SELECT statement.

For example, if you are retrieving information about employees, you can sort on department, and then within each department, you can sort on employee ID.

**❖ To define ORDER BY criteria:**

- 1 Click the Sort tab in the SQL Toolbox.

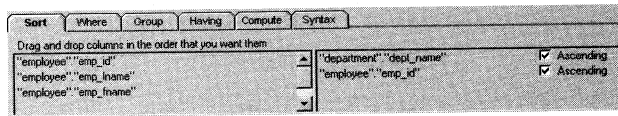
The columns you selected display in the order of selection. You may need to scroll to see your selections.

- 2 Drag the first column you want to sort on to the right side of the Sort tab.

This specifies the column for the first level of sorting. By default, the column is sorted in ascending order. To specify descending order, clear the Ascending checkbox.

- 3 Continue to specify additional columns for sorting in ascending or descending order as needed.

You can change the sorting order by dragging the selected column names up or down. With the following sorting specification, rows will be sorted first by department name, then by employee ID:



- 4 Define limiting (Where tab), grouping (Group tab), and limiting groups (Having tab) criteria as appropriate.
- 5 Click the SQL Select button to return to the Report painter workspace.

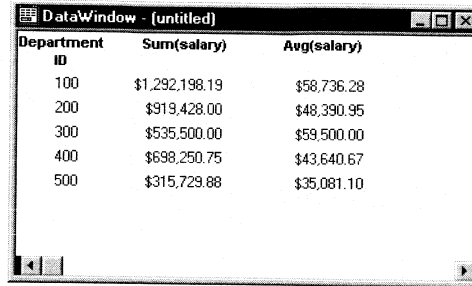
#### Defining GROUP BY criteria

You can group the retrieved rows by specifying groups that correspond to the GROUP BY clause in the SELECT statement. This grouping happens *before* the data is retrieved into the report. Each group is retrieved as one row into the report.

For example, if in the SELECT statement you group data from the Employee table by department ID, you will get one row back from the database for every department represented in the Employee table. You can also specify computed columns, such as total and average salary, for the grouped data. Here is the corresponding SELECT statement:

```
SELECT dept_id, sum(salary), avg(salary)
FROM employee
GROUP BY dept_id
```

If you specify this with the Employee table in the Powersoft Demo Database, you will get five rows back, one for each department:



Department ID	Sum(salary)	Avg(salary)
100	\$1,292,198.19	\$58,736.28
200	\$919,428.00	\$48,390.95
300	\$535,500.00	\$59,500.00
400	\$698,250.75	\$43,640.67
500	\$315,729.88	\$35,081.10

FOR INFO For more about GROUP BY, see your DBMS documentation.

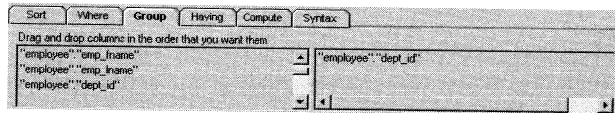
#### ❖ To define GROUP BY criteria:

- 1 Click the Group tab in the SQL Toolbox.

The columns in the tables you selected display in the left side of the Group tab. You may need to scroll to see your selections.

- 2 Drag the first column you want to group on to the right side of the Group tab.

This specifies the column for grouping. Columns are grouped in the order they are displayed in the right side of the Group tab. In the following, the report will be grouped by department ID:



- 3 Continue to specify additional columns for grouping within the first grouping column as needed.

To change the grouping order, drag the column names in the right side to the positions you want.

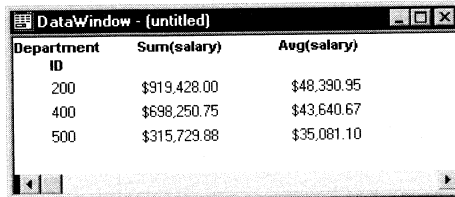
- 4 Define sorting (Sort tab), limiting (Where tab), and limiting groups (Having tab) criteria as appropriate.
- 5 Click the SQL Select button to return to the DataWindow painter workspace.

#### Defining HAVING criteria

If you have defined groups, you can define HAVING criteria to restrict the retrieved groups. For example, if you group employees by department, you can restrict the retrieved groups to departments whose employees have an average salary of less than \$50,000. This corresponds to:

```
SELECT dept_id, sum(salary), avg(salary)
FROM employee
GROUP BY dept_id
HAVING avg(salary) < 50000
```

If you specify this with the Employee table in the Powersoft Demo Database, you will get three rows back, because there are three departments that have average salaries less than \$50,000:



Department ID	Sum(salary)	Avg(salary)
200	\$919,428.00	\$48,390.95
400	\$698,250.75	\$43,640.67
500	\$315,729.88	\$35,081.10

❖ **To define HAVING criteria:**

- ◆ Click the Having tab in the bottom of the workspace.

Each row in the Having tab is a place for entering an expression that limits which groups are retrieved. For information on how to define criteria on the Having tab, see the procedure in "Defining WHERE criteria" on page 176 .

## Using Query

When you choose Query as the data source, you select a predefined SQL SELECT statement (a **query**) as specifying the data for your report.

❖ **To define the data using Query:**

- 1 Click Query in the New Report dialog box, select a presentation style, and click OK.

The Select Query dialog box displays, listing all queries that have been defined in the current InfoMaker library.

- 2 Select a query and click OK.

The Report painter workspace displays (with some presentation styles, you need to provide additional information before going to the workspace).

FOR INFO To learn how to create queries, see "Defining queries" on page 188.



## Using External

If the data for the report is not coming from a database (either through a native Powersoft database interface or through ODBC), specify External as the data source. You then specify the data columns and their types so InfoMaker can build the appropriate report to hold the data. These columns make up the **result set**. InfoMaker places the columns you specified in the result set in the report.

---

### Using ODBC drivers instead of External

If you installed ODBC drivers when you installed InfoMaker, you have an ODBC dBASE driver and an ODBC text driver on your computer.

If you configure ODBC data sources for the text file driver and the dBASE driver, you can access data that resides in a dBASE file or a tab-separated text file. To configure data sources, click the ODBC button in the PowerBar.

Accessing data by means of the ODBC drivers is easier than using External as your InfoMaker data source, because External requires you to specify your data as a result set. In other words, to use External you have to *know* your data.

**FOR INFO** For information on ODBC drivers, see *Connecting to Your Database*.

---

#### ❖ To define the data using External:

- 1 Click External in the New Report dialog box, select a presentation style, and click OK.

The Result Set Description dialog box displays for you to specify the first column in the result set.

- 2 Enter the name and type of the column.  
Available data types are listed in the dropdown listbox.
- 3 Click Add to enter the name and type of any additional columns you want in the result set.
- 4 Click OK when you have added all the columns you want.

What you do next

Now you must import the data values from the file into the report. This is similar to retrieving data from the database.

#### ❖ To import the data values from an external file:

- 1 Click the Preview button in the PainterBar.  
If the button isn't visible, turn toolbar text off.

- 2 Select Rows>Import from the menu bar.  
The Select Import File dialog box displays.
- 3 Select the type of files to list from the List Files of Type dropdown listbox (either TXT or DBF files).
- 4 Enter the name of the import file and click OK.  
Alternatively, you can select the name from the file list. Use the Drives dropdown listbox and the Directories box as needed to display the list of files that includes the one you want.

## Using Stored Procedure

A **stored procedure** is a set of precompiled and preoptimized SQL statements that performs some database operation. Stored procedures reside where the database resides, and you can access them as needed.

You can specify a stored procedure as the data source for a report if your DBMS supports stored procedures.

**FOR INFO** For information on support for stored procedures, see your database documentation.

---

### **If the Stored Procedure icon is not displayed**

The icon for the Stored Procedure data source displays in the New Report dialog box only if the database to which you are connected supports stored procedures.

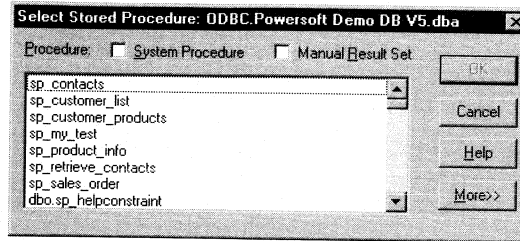
---

#### ❖ **To define the data using Stored Procedure:**

- 1 Select Stored Procedure in the New Report dialog box.

- 2 Select a presentation style and click OK.

The Select Stored Procedure dialog box displays a list of the stored procedures in the current database:



- 3 Select a stored procedure from the list.

To list system procedures, select the System Procedure checkbox.

To see the syntax of the selected stored procedure, click More. If this is not the stored procedure you want, select a different one.

- 4 Specify how you want the result set description built:
  - ◆ **Build the result set description automatically** Clear the Manual Result Set checkbox.

InfoMaker executes the stored procedure and builds the result set description for you. You go to the Report painter workspace with the columns placed in the report.

- ◆ **Define the result set description manually** Select the Manual Result Set checkbox and click OK.

In the Result Set Description dialog box:

- ◆ Enter the name and type of the first column in the result set.
- ◆ To Add additional columns, click Add.
- ◆ To define retrieval arguments or change to another stored procedure, click More.

---

#### Your preference is saved

InfoMaker records your preference for building result set descriptions for stored procedure reports in the variable `Stored_Procedure_Build` in the InfoMaker initialization file. If this variable is set to 1, InfoMaker will automatically build the result set; if the variable is set to 0, you will be prompted to define the result set description.

---

- 5 When you have defined the entire result set, click OK.

You go to the Report painter workspace with the columns specified in the result set placed in the report.

FOR INFO For information about defining retrieval arguments for reports, see Chapter 5, "Enhancing Reports".

❖ **To edit the result set description:**

- 1 Click SQL Select in the PainterBar.

*or*

Select Design>Edit Data Source from the menu bar.

- 2 Click More to edit the Execute statement, select another stored procedure, or add arguments.

- 3 When you have defined the entire result set, click OK.

You return to the Report painter workspace with the columns specified in the result set placed in the report.

FOR INFO For information about defining retrieval arguments for reports, see Chapter 5, "Enhancing Reports".

## Generating and saving a report

Once you have selected a presentation style and data source, InfoMaker generates the report and takes you to the Report painter workspace (with some presentation styles, you are prompted for additional information before you go to the workspace).

When generating the report, InfoMaker may use information from a facility called the **Powersoft repository**. If repository information is available, InfoMaker uses it.

### About the Powersoft repository and reports

The Powersoft repository is a set of system tables maintained by the Database painter. It contains information about database tables and columns. Repository information extends database definitions by recording information that is relevant to using database data in screens and reports.

---

#### **Repository information applies to forms, too**

InfoMaker uses repository information when generating a form the same way it uses it when generating a report.

---

For example, labels and headings you defined for columns in the Database painter are used in the generated report. Similarly, if you associated an edit style with a column in the Database painter, that edit style is automatically used for the column in the report.

When generating a report, InfoMaker uses the following information from the repository:

<b>For</b>	<b>InfoMaker uses</b>
Tables	Fonts specified for labels, headings, and data
Columns	Text specified for labels and headings Display formats Column widths, heights, and justifications Edit styles

If there is no repository information for the database tables and columns you are using, you can set the text for headings and labels, the fonts, and the display formats in the Report painter workspace. The difference is that you have to do this individually for every report that you create using the data.

If you want to change something that came from the repository, you can change it in the Report painter workspace. The changes you make in the Report painter apply only to the report you are working on.

The advantage of the repository is that it saves time and ensures consistency. You only have to specify the information once, in the database. Since InfoMaker uses the information whenever anyone creates a new report with the data, it's more likely that the appearance and names of data items will be consistent.

---

**If you have installed InfoMaker without the Database painter**

You cannot create repository information if you have installed InfoMaker without the Database painter. Another InfoMaker user with the Database painter or a PowerBuilder user can add repository information into the database for you. Or you can change the extended attributes the repository handles individually in each report.

---

**FOR INFO** For more information about the repository, see Chapter 3, "Managing the Database", and the Appendix, "The Powersoft Repository".

## Saving the report

When you have created a report and are in the Report painter workspace, you should save the report. The first time you save it you will give it a name. As you work, you should save your report frequently so that you don't lose changes.

❖ **To save the report:**

- 1 Select File>Save from the menu bar.

If you have previously saved the report, InfoMaker saves the new version in the same library and returns you to the Report painter workspace.

If you have not previously saved the report, InfoMaker displays the Save Report dialog box.

- 2 (Optional) Enter comments in the Comments box to describe the report.
- 3 Enter a name for the report in the Reports box.
- 4 Click OK.

## **Naming the report**

The report name can be any valid InfoMaker identifier up to 40 contiguous characters.

**FOR INFO** For information about InfoMaker identifiers, see Chapter 21, "Identifiers".

## Defining queries

A query is a SQL SELECT statement created with the Query painter and saved with a name so that it can be used repeatedly as the data source for a report.

Queries save time, because you specify all the data requirements just once. For example, you can specify the columns, which rows to retrieve, and the sorting order in a query. Whenever you want to create a report using that data, simply specify the query as the data source.

### ❖ To define a query:

- 1 Click the Query button in the PowerBar.
- 2 Click New in the Select Query dialog box to define a new query.
- 3 Select tables and columns in the Select Tables dialog box.

You can also define sorting and grouping criteria, define computed columns, and so on, exactly as you do when creating a report using the SQL Select data source.

**FOR INFO** For more about defining the SELECT statement, see "Using SQL Select" on page 166.

## Previewing the query

While creating a query, you can preview it to make sure it is retrieving the correct rows and columns.

### ❖ To preview a query:

- 1 In the Query painter, click the Preview button.  
*or*  
Select Design>Preview from the menu bar.

InfoMaker retrieves the rows satisfying the currently defined query in a grid-style report.

- 2 Manipulate the retrieved data as you do in the Data Manipulation painter.

You can sort and filter the data, but you cannot insert or delete a row or apply changes to the database.

**FOR INFO** For more about the Data Manipulation painter, see Chapter 3, "Managing the Database".



- 3 When you have finished previewing the query, click the Preview button to return to the Query painter workspace.

## Saving the query

### ❖ To save a query:

- 1 Select File>Save from the menu bar.

*or*

Click the Save button.

If you have previously saved the query, InfoMaker saves the new version and returns you to the Query painter workspace.

If you have not previously saved the query, InfoMaker displays the Save Query dialog box.

- 2 Enter a name for the query in the Queries box (see "Naming the query" next).

- 3 (Optional) Enter comments to describe the query.

These comments display in the Environment painter. It is a good idea to use comments to remind yourself and others of the purpose of the query.

- 4 Click OK.

## Naming the query

The query name can be any valid InfoMaker identifier up to 40 characters.

**FOR INFO** For information about InfoMaker identifiers, see Chapter 21, "Identifiers".

### A recommendation

When you name queries, use a unique name to identify each one. One approach is a 2-part name: a standard prefix that identifies the object as a query (such as q\_) and a unique suffix.

For example, you might name a query that displays employee data q\_emp\_data.

## Modifying a query

❖ **To modify a query:**

- 1 Click the Query button in the PowerBar.  
The Select Query dialog box displays.
- 2 Select the query you want to modify and click Design.
- 3 Modify the query as needed.

## **What's next**

After you have generated your report, you will probably want to preview it to see how it looks. After that, you may want to enhance the report in the Report painter workspace before using it. InfoMaker provides many ways for you to make a report easier to use and more informative.

See Chapter 5, "Enhancing Reports" next.



# Enhancing Reports

## About this chapter

After InfoMaker generates the basic report, you can further enhance its appearance and content. You do that in the Report painter workspace. This chapter describes basic enhancements you can make to a report.

## Contents

Topic	Page
Working in the Report painter workspace	195
Previewing a report	205
Modifying general report properties	220
Reorganizing objects in a report	235
Prompting for retrieval criteria in a report	242
Adding objects to a report	244
Positioning objects in a report	256
Storing data in a report	257
Retrieving rows as needed	259
Saving retrieved rows to disk	260

## Related topics

Other ways to enhance reports are covered in later chapters:

Chapter	Explains how to
Chapter 6, "Displaying and Validating Data"	Specify display formats, edit styles, and validation rules for column data
Chapter 7, "Filtering, Sorting, and Grouping Rows"	Limit which rows are displayed, the order in which they are displayed, and whether they are divided into groups
Chapter 8, "Highlighting Information in Reports and Forms"	Highlight data by using conditional expressions to modify the properties of objects in reports
Chapter 9, "Using Nested Reports"	Place reports inside reports

<b>Chapter</b>	<b>Explains how to</b>
Chapter 10, "Working with Graphs"	Use graphs to visually present information retrieved in a report
Chapter 11, "Working with Crosstabs"	Use crosstabs to present analyses of data retrieved in a report

## Working in the Report painter workspace

Once you have specified your presentation style and data source, InfoMaker generates a basic report and places you in the Report painter workspace where you can enhance the report.

This section presents an overview of working in the workspace:

- ◆ "Understanding the Report painter workspace" next
- ◆ "Using the Report painter toolbars" on page 198
- ◆ "Using property sheets in the Report painter" on page 199
- ◆ "Selecting objects in the Report painter" on page 200
- ◆ "Using keyboard shortcuts in the Report painter" on page 202
- ◆ "Resizing bands in the Report painter workspace" on page 204
- ◆ "Using zoom in the Report painter" on page 204
- ◆ "Undoing changes in the Report painter" on page 204

---

### Applicability of information

Some of the information presented in this section doesn't apply to all the report presentation styles.

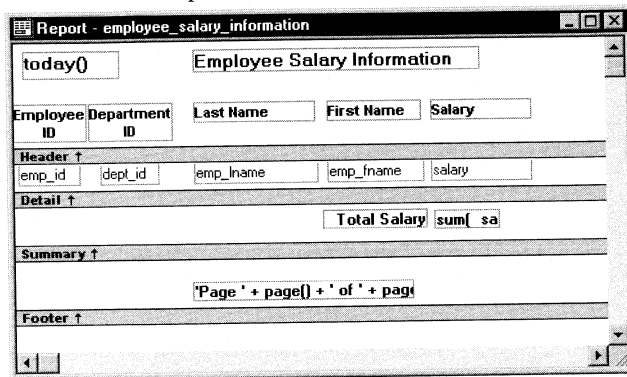
---

## Understanding the Report painter workspace

With most presentation styles, the Report painter workspace is divided into areas called **bands**. Each band corresponds to a section of the displayed report.

A report is divided into four bands: header, detail, summary, and footer. Each band is identified by a bar containing the name of the band above the bar and an arrow pointing to the band.

Here is the workspace for a tabular report:



Band	Used to display
Header	Information at the top of every screen or page, such as the name of the report or current date
Detail	Data from the database or other data source
Summary	Summary information that displays after all the data, such as totals and counts
Footer	Information displayed at the bottom of every page or screen, such as page number and page count

## About the header band

The header band contains heading information that is displayed at the top of every screen or page.

When InfoMaker generates the basic report, the presentation style determines the contents of the header band:

- ◆ If the presentation style is Tabular, Grid, or N-Up, the headings defined for the columns in the Database painter display in the header band and the columns display on a single line across the detail band
- ◆ If the presentation style is Freeform, the header band is empty and labels display in the detail band next to each column

You can specify additional heading information (such as a date) in the header band and can include pictures, graphic objects, and color to enhance the appearance of the band.



---

**Displaying the current date**

To include the current date in the header, you place a computed field that uses the Today InfoMaker function in the header band.

FOR INFO For information, see "Adding computed fields to a report" on page 248.

---

**About the detail band**

The detail band displays the retrieved data. The number of rows of data that display in the report at one time is determined by the following expression:

$$\frac{(\text{Height of the report} - \text{Height of headers and footers})}{\text{Height of the detail band}}$$

When InfoMaker generates the basic report, the presentation style determines the contents of the detail band:

- ◆ If the presentation style is Tabular, Grid, N-Up, or Label, the detail band displays column names, representing the columns
- ◆ If the presentation style is Freeform, the labels defined for the columns in the Database painter display in the detail band with boxes for the data to the right

---

**How InfoMaker names the columns in the workspace**

If the report uses one table, the names of the columns in the workspace are the same as the names in the table.

If the report uses more than one table, the names of the columns in the workspace are *tablename\_columnname*. InfoMaker prefaces the name of the column with the table name to prevent ambiguity, since different tables can have columns with the same name.

---

When you design the detail band of a report, you can specify display information for each column of the report and add other objects, such as text, pictures, drawing objects, and graphs.

## About the summary and footer bands

You use the summary and footer bands of the report the same way you use summary pages and page footers in a printed report:

- ◆ The contents of the summary band display at the end, after all the detail rows; this band often summarizes information in the report.
- ◆ The contents of the footer band display at the bottom of each screen or page of the report; this band often displays the page number and name of the report.

These bands can contain any information you want, including text, drawing objects, graphs, and computed fields containing aggregate totals.

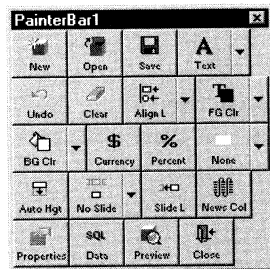
## Using the Report painter toolbars

The Report painter contains a customizable PainterBar and a StyleBar, which are described in this section.

**FOR INFO** For more information about using toolbars, see "Using toolbars" on page 24.

### PainterBar

The PainterBar includes buttons for standard operations (such as Open, Save, Undo) and for common reporting operations (such as Preview, Percent). It also includes six dropdown toolbars, which are indicated by a small black triangle on the right part of a button:



### Dropdown toolbars

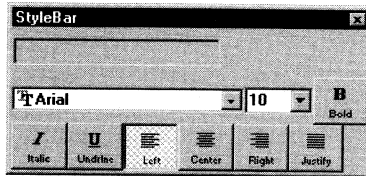
These are the dropdown toolbars that are available:

Dropdown Toolbar	Used to
Background Color	Specify the background color of one or more selected objects
Borders	Specify borders for one or more selected objects

Dropdown Toolbar	Used to
Foreground Color	Specify the foreground color of one or more selected objects. In a text object, the foreground color specifies the color of the text
Layout	Specify the alignment, sizing, and spacing of selected objects
Objects	Specify objects to add to a report
Slide	Specify sliding for objects

### StyleBar

The StyleBar includes buttons for applying properties (such as font size and bold) to selected text elements:



## Using property sheets in the Report painter

Each part of the report (such as text, columns, computed fields, bands, graphs, even the report itself) has a property sheet. The items in the property sheet are appropriate to the part.

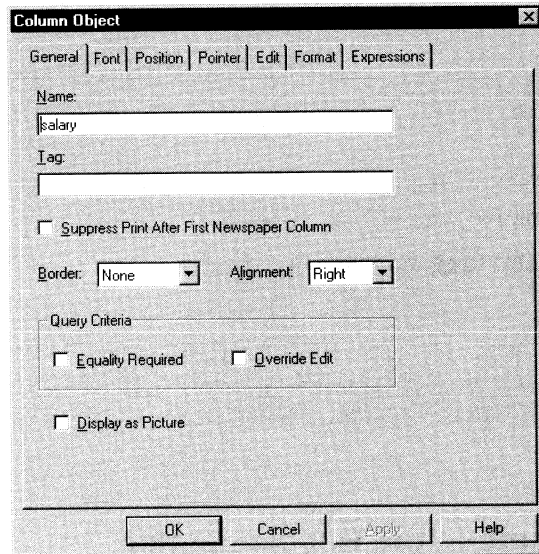
You can use property sheets to modify the parts of the report.

❖ **To use a property sheet in the Report painter workspace:**

- 1 Position the mouse over the part you want to modify.
- 2 Display the part's popup menu and select Properties.

The appropriate property sheet displays.

The following property sheet is for a column. The Column property sheet for a report has seven tabbed property pages of information, which you access by clicking the appropriate tab. For example, if you want to choose an edit style for the column, you click the Edit tab. This brings the Edit property page to the front of the property sheet:



When you want to modify the properties of part of a report, display the property sheet and pick the properties you want to change. Click the various tabs to change property pages.

## Selecting objects in the Report painter

The Report painter provides several ways to select objects to act on. You can select multiple objects and act on all the selected objects as a unit. For example, you can move all of them or change the fonts used to display text for all of them.

---

### Lasso selection

Use **lasso selection** when possible because it's fast and easy. Lasso selection is another name for the method described below for selecting neighboring multiple objects.

---

❖ **To select one object in a report:**

- ◆ Click it.

The object displays with handles on it. Previously selected objects are no longer selected.

❖ **To select neighboring multiple objects in a report (lasso selection):**

- 1 Press and hold the left mouse button (mouse button on the Macintosh) at one corner of the neighboring objects.
- 2 Drag the mouse over the objects you want to select.  
InfoMaker displays a bounding box (the lasso).
- 3 Release the mouse button.  
All the objects in the bounding box are selected.

❖ **To select non-neighboring multiple objects in a report:**

- 1 Click the first object.
- 2 Press and hold the CTRL key and click additional objects.

---

**On Macintosh**

On the Macintosh, press the COMMAND key instead of the CTRL key.

---

All the objects are selected.

❖ **To select objects by type or position in the report:**

- ◆ Do the following:

To select	Select this item from the menu bar	Or press this shortcut key
All objects	Edit>Select>Select All	CTRL+A
All text	Edit>Select>Select Text	
All columns	Edit>Select>Select Columns	
Objects in relation to the currently selected object	Edit>Select>Select Above Edit>Select>Select Below Edit>Select>Select Left Edit>Select>Select Right	CTRL+UP ARROW CTRL+DOWN ARROW CTRL+LEFT ARROW CTRL+RIGHT ARROW

---

**On Macintosh**

On the Macintosh, press the COMMAND key instead of the CTRL key.

---

## Displaying information about the selected object

The name, x and y coordinates, width, and height of the selected object are displayed in the MicroHelp bar. If multiple objects are selected, *Group Selected* displays in the Name area and the coordinates and size do not display.

## Using keyboard shortcuts in the Report painter

The following table lists the keyboard shortcuts available in the Report painter.

---

**On Macintosh**

On the Macintosh, press the COMMAND key instead of the CTRL key and the OPTION key instead of the ALT key.

---

Action	Key combination	Comments
Boldface text	CTRL+B	Toggles boldface on and off for all selected objects
Center text	CTRL+SHIFT+C	
Clear	DEL	Clears all selected objects
Close painter	CTRL+F4 CTRL+W	
Close InfoMaker	ALT+F4 CTRL+Q	Available everywhere on Windows. Not available on Macintosh.
Copy	CTRL+C	Copies objects
Cut	CTRL+X	Cuts objects
Edit text	CTRL+SHIFT+E	Activates the Text box in the StyleBar
File editor	SHIFT+F6	Available everywhere
Italicize text	CTRL+I	Toggles italic on and off for all selected objects
Left justify text	CTRL+SHIFT+L	

<b>Action</b>	<b>Key combination</b>	<b>Comments</b>
Move object	ARROW	
New report	CTRL+N	Creates a new report
Open report	CTRL+O	Opens an existing report
Paste	CTRL+V	Paste objects
PowerPanel	CTRL+SHIFT+N	Available everywhere
Preview	CTRL+SHIFT+P	Toggles between preview and the design workspace
Print	CTRL+P	
Resize object	SHIFT+ARROW	
Return to the object	CTRL+SHIFT+O	Returns focus to the object from the StyleBar
Right justify text	CTRL+SHIFT+R	
Save report	CTRL+S	
Select above	CTRL+UP ARROW	Selects all objects above the currently selected object
Select all	CTRL+A	Selects all objects in the report
Select below	CTRL+DOWN ARROW	Selects all objects below the currently selected object
Select left	CTRL+LEFT ARROW	Selects all objects that are left of the currently selected object
Select right	CTRL+RIGHT ARROW	Selects all objects that are right of the currently selected object
Switch to	ALT+ESC	Available everywhere
Tab	TAB	Tabs to next object
Tab backwards	SHIFT+TAB	Tabs to previous object
Underline text	CTRL+SHIFT+U	Toggles underline on and off for all selected objects
Undo/Redo	CTRL+Z	Undoes the most recent change (including the most recent undo)

## Resizing bands in the Report painter workspace

You can change the size of any band in the report.

❖ **To resize a band in the Report painter workspace:**

- ◆ Position the pointer on the bar representing the band and drag the bar up or down to shrink or enlarge the band.

## Using zoom in the Report painter

You can zoom the display in and out so you can get a better idea of how your report looks. For example, if you are working with a large report, you can zoom out so you can see all of it on your screen. Or you can zoom in on a group of objects to better see their details.

❖ **To zoom the display in the Report painter:**

- 1 Select Design>Options from the menu bar.

The Report Options property sheet displays.

- 2 Select the Zoom property page.
- 3 Select a built-in zoom percentage.

*or*

Set a custom zoom percentage by typing an integer in the Custom box.

## Undoing changes in the Report painter

You can undo your most recent change in the workspace by pressing CTRL+Z or selecting Edit>Undo from the menu bar. (If you have just undone an action, the menu item changes to Edit>Redo.)

---

### **On Macintosh**

On the Macintosh, press the COMMAND key instead of the CTRL key.

---



## Previewing a report

You can preview a report to view it as it will appear and test the processing that takes place in it. This section provides information on what you can do while previewing a report, including:

- ◆ "Retrieving data" on page 206
- ◆ "Sorting and filtering data" on page 209
- ◆ "Importing data into a report" on page 209
- ◆ "Using preview" on page 210
- ◆ "Printing data" on page 211
- ◆ "Saving data in an external file" on page 212
- ◆ "Saving the data in HTML Table format" on page 213
- ◆ "Working with PSR files" on page 215
- ◆ "Mailing reports" on page 217
- ◆ "Working in a grid report" on page 218

Changes you make to the report while previewing are maintained when you return to the workspace. For example, if you specify sorting while previewing, the sorting becomes part of the design of the report.

### ❖ To preview the report:

- 1 Click the Preview button in the PainterBar.  
*or*  
Select Design>Preview from the menu bar.

You are now in preview. The bars that indicate the bands disappear, and by default InfoMaker retrieves all the rows from the database (you are prompted to supply arguments if you defined retrieval arguments).

---

### In external reports

If the report uses the External data source, no data is retrieved. You can import data, as described in "Importing data into a report" on page 209.

---

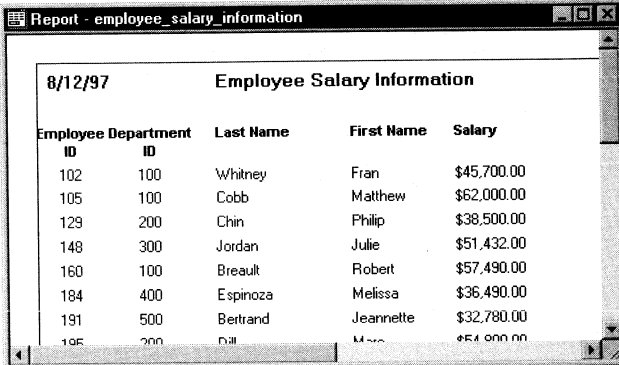
---

### In reports that have stored data

If the report has stored data in it, no data is retrieved from the database.

---

As the rows are being retrieved, the Retrieve button changes to a Cancel button. You can click the Cancel button to stop the retrieval:



The screenshot shows a window titled "Report - employee\_salary\_information" with a table of employee salary data. The table has columns for Employee ID, Department ID, Last Name, First Name, and Salary. The data is as follows:

Employee ID	Department ID	Last Name	First Name	Salary
102	100	Whitney	Fran	\$45,700.00
105	100	Cobb	Mathew	\$62,000.00
129	200	Chin	Philip	\$38,500.00
148	300	Jordan	Julie	\$51,432.00
160	100	Breault	Robert	\$57,490.00
184	400	Espinoza	Melissa	\$36,490.00
191	500	Bertrand	Jeannette	\$32,780.00
195	200	Dill	Marc	\$44,000.00

- 2 Evaluate your report.

For example, scroll the page and page down to new pages to review the layout and content of the report.

- 3 To leave preview, click the Preview button.

*or*

Select Design>Preview from the menu bar.

You return to the workspace.

## Retrieving data

Where InfoMaker gets data

When you preview a report, InfoMaker follows this order of precedence to supply the data in your report:

- 1 If you have saved data in the report (using Rows>Data from the menu bar), InfoMaker uses the saved rows from the report and does not retrieve data from the database.
- 2 If data caching is on, InfoMaker uses the data in the cache (if there is any). Data caching is on by default.
- 3 If there is no data in the cache yet or if data caching is not on, InfoMaker retrieves data from the database automatically, with one exception. If the Retrieve on Preview option is off, you have to request retrieval explicitly, as described next.

Previewing without  
retrieving data

If you do not want InfoMaker to retrieve data from the database automatically when you first preview, you can set the Retrieve on Preview option. Then you can preview the report without retrieving data.

❖ **To be able to preview without retrieving data automatically:**

- 1 With the report displayed in the workspace, select Design>Options from the menu bar.

The Report Options property sheet displays.

- 2 Select the General tab.

- 3 Deselect the Retrieve on Preview checkbox.

When this checkbox is unchecked, your request to preview the report will not result in automatic data retrieval from the database.

InfoMaker uses data  
caching

By default, when InfoMaker first retrieves data during preview, it stores the data internally. If you go back to the workspace and then return to preview, InfoMaker displays the stored data instead of retrieving rows from the database again. This can save you a lot of time since data retrieval can be time consuming.

---

**On Windows 3.1, SHARE must be loaded**

In order for InfoMaker under Windows 3.1 to cache the data, you must be running the DOS SHARE program. If you use Windows for Workgroups, Windows NT, or Windows 95, SHARE is built in; do not run the DOS SHARE program.

---



---

**On Macintosh**

On Macintosh systems, InfoMaker does not cache data when you repeatedly preview a report. Caching data for display during execution using the Rows>Data command is a separate function and works the same way on Macintosh as it does on Windows.

---

How using data from  
the cache affects you

Because InfoMaker accesses the cache and does not automatically retrieve data every time you preview, you may not have what you want when you preview. The data you see in preview and the data in the database can be out of sync.

For example, if you are working with live data that changes frequently or with statistics based on changing data, and you spend time designing the report, the data you are looking at may no longer match the database. In this case, you may want to retrieve again just before printing.

Turning off data caching

If you do not want InfoMaker to use data from the cache, you can turn off the caching of retrieved data.

❖ **To turn off data caching so that InfoMaker will retrieve fresh data from the database every time you preview a report:**

- 1 With the report displayed in the workspace, select Design>Options from the menu bar.

The Report Options property sheet displays.

- 2 Select the General tab.
- 3 Deselect the Retain Data to Design checkbox.

From now on, InfoMaker will go to the database and not to the cache for data.

Explicitly retrieving data

You can explicitly request retrieval anytime.

❖ **To retrieve rows from the database:**

- ◆ Click the Retrieve button in the PainterBar.  
*or*  
Select Rows>Retrieve from the menu bar.

---

**If the report has retrieval arguments or is set up to prompt for criteria**

You will be prompted to supply values for the arguments or to specify criteria.

---

InfoMaker retrieves the rows. As InfoMaker retrieves, the Retrieve button changes to a Cancel button. You can click the Cancel button to stop the retrieval anytime.

Other options that affect retrieval

These other options can affect retrieval:

- ◆ **Query Governor** Lets you (or a database administrator) set certain restrictions on data retrieval. For example, Query Governor can limit the number of rows that will be retrieved.  
  
FOR INFO For information, see "Query Governor" on page 61.
- ◆ **Retrieve Rows As Needed** Lets you specify that only the rows needed to display the current portion of the report should be retrieved. When you scroll downward, additional rows are retrieved. This can speed up reporting in certain situations.

FOR INFO See "Retrieving rows as needed" on page 259.

- ◆ **Retrieve Rows to Disk** (Windows 95 and Windows NT only) Lets you specify that InfoMaker should save retrieved data on your hard disk in a temporary file rather than keep the data in memory. When you preview your report, InfoMaker swaps rows of data from the temporary file into memory as needed.

FOR INFO For information, see "Saving retrieved rows to disk" on page 260.

## Sorting and filtering data

You can define and use sort criteria and filters for the rows in preview. The sort criteria and filters you define while previewing are maintained when you return to the workspace.

FOR INFO See Chapter 7, "Filtering, Sorting, and Grouping Rows".

## Importing data into a report

While previewing, you can import and display data from an external source. Then you can view the data or save the data in an external format.

---

### **Importing is not available in a report packaged in an executable file**

When you create an InfoMaker application (which is an executable file), you can include reports. When you run a report from the executable file, you cannot import data.

---

#### ❖ **To import data into a report while previewing:**

- 1 Select Rows>Import from the menu bar.

The Select Import File dialog box displays.

- 2 Specify the file from which you want to import the data.

The types of files that you can import into the painter display in the List Files of Type dropdown listbox.

- 3 Click Open.

InfoMaker reads the data from the file into the painter. You can view the data and save the data in an external file.

---

### Data from file must match retrieved columns

When importing data from a file, the data must match all the columns in the retrieved data (typically, the columns specified in the SELECT statement), not just the columns that are displayed in the report.

---

## Using preview

You must be previewing to print reports (including standalone graphs and crosstabs). Preview displays the report as it will print.

You can use rulers and zooming to help you review the report online. When you are satisfied with the report, you define print specifications and then print.

---

### Using the IntelliMouse pointing device

Using the IntelliMouse pointing device, you can scroll a report (at execution time or in preview) by rotating the wheel. You can also zoom a report larger or smaller by holding down the CTRL key while rotating the wheel.

---

Controlling the display of rulers

You can choose whether to display rulers around page borders.

❖ **To control the display of rulers:**

- ◆ Select/deselect File>Print Preview Rulers from the menu bar.

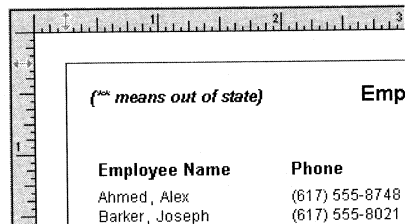
Changing margins

You can dynamically change margins while previewing a report.

❖ **To change the margins while previewing:**

- ◆ Drag the margin boundaries on the rulers.

The following picture shows the left and top margin boundaries. There are also boundaries for the right and bottom margins:



### Zooming the page

You can reduce or enlarge the amount of the page that displays on the screen. This does not affect the printed output.

❖ **To zoom the page on the display screen:**

- 1 Select File>Print Preview Zoom from the menu bar.

The Zoom dialog box displays.

- 2 Select the magnification you want and click OK.

The display of the page zooms in or out as appropriate. The size of the contents of the page changes proportionately as you zoom. This type of zooming affects your display but does not affect printing.

### Zooming the contents

In addition to zooming the display on the screen, you can also zoom the contents, affecting the amount of material that prints on a page.

❖ **To zoom the contents of a report with respect to the printed page:**

- 1 Select Design>Zoom from the menu bar.

The Zoom dialog box displays.

- 2 Select the magnification you want and click OK.

The contents of the page zooms in or out as appropriate. If you enlarge the contents so they no longer fit, InfoMaker creates additional pages as needed.

## Printing data

You can print your report while previewing. You can print all pages, a range of pages, only the current page, or only odd or even pages. You can also specify whether you want multiple copies, collated copies, and printing to a file.

---

### **To change printers or settings before printing**

You can choose File>Printer Setup from the menu bar in preview. On the Macintosh, you must set the printer specifications before printing.

---

❖ **To print:**

- 1 Select File>Print from the menu bar to display the Print dialog box.
- 2 Specify the number of copies to print.
- 3 Specify the pages: select All or Current Page, or type page numbers and/or page ranges in the Pages box.

- 4 Specify all pages, even pages, or odd pages in the Print dropdown listbox.
- 5 If you want to print to a file rather than to the printer, select the Print to File checkbox.
- 6 If you want to change the collating option, deselect or select the Collate Copies checkbox and click OK.

If you specified print to file, the Print to File dialog box displays.

- 7 Enter a filename and click OK.

The extension PRN identifies it as a file prepared for the printer. Change drive and/or directory if you want.

## Saving data in an external file

While previewing, you can save the data retrieved in an external file. Note that the data and headers (if specified) are saved. Information in the footer or summary bands is not saved unless you are saving in the file format named Powersoft Report.

**FOR INFO** For more information about the Powersoft Report file format, see "Working with PSR files" on page 215.

### ❖ To save the data in a report in an external file:

- 1 Select File>Save Rows As from the menu bar.

The Save As dialog box displays.

- 2 Choose a format for the file from the Save As Type dropdown listbox.

If you want the column headers saved in the file, select a file format that includes headers (such as Excel With Headers). When you select a *with headers* format, the names of the database columns (not the column labels) will also be saved in the file.

When you choose a format, InfoMaker supplies the appropriate file extension on platforms that require them.

---

### **Saving the data as a Powersoft report or HTML Table**

You can choose Powersoft Report in the Save As dialog box to save a Powersoft PSR file. You can choose HTML Table in the Save As dialog box to save the data in HTML Table format.

---



- 3 Name the file and click Save.

InfoMaker saves all displayed rows in the file; all columns in the displayed rows are saved. Filtered rows are not saved.

## **Saving the data in HTML Table format**

One of the formats you can choose to save data in is HTML Table format. When you save in HTML Table format, InfoMaker saves a style sheet along with the data. If you use this format, you can open the saved file in a browser such as Netscape. Once you have the file in HTML Table format, you can continue to enhance the file in HTML.

### About the results

Some presentation styles translate into better HTML than others. The Tabular, Group, Freeform, Crosstab, and Grid presentation styles produce good results. The Composite, RichText, Graph, and OLE 2.0 presentation styles and nested reports produce HTML tables based on the result set (data) only and not on the presentation style. Reports with overlapping objects in them may not produce the desired results.

### An example

The example shows a report in preview and the file saved in HTML Table format in Netscape.

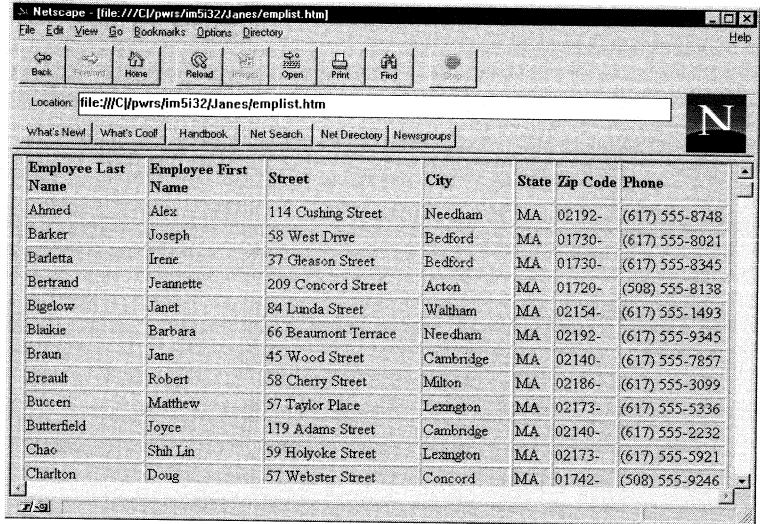
These are the steps you would follow to recreate this example:

- 1 Open a report.
- 2 Preview it:

Employee List							Page 1 of 3
5/8/96							
Employee Last Name	Employee First Name	Street	City	State	Zip Code	Phone	
Ahmed	Alex	114 Cushing Street	Needham	MA	02192-	(617) 555-8748	
Barker	Joseph	58 West Drive	Bedford	MA	01730-	(617) 555-8021	
Barletta	Irene	37 Gleason Street	Bedford	MA	01730-	(617) 555-8345	
Bertrand	Jeannette	209 Concord Street	Acton	MA	01720-	(508) 555-8138	
Bigelow	Janet	84 Lunda Street	Waltham	MA	02154-	(617) 555-1493	
Blaikie	Barbara	66 Beaumont Terrace	Needham	MA	02192-	(617) 555-9345	
Braun	Jane	45 Wood Street	Cambridge	MA	02140-	(617) 555-7857	
Breault	Robert	58 Cherry Street	Milton	MA	02186-	(617) 555-3099	
Bucceri	Matthew	57 Taylor Place	Lexington	MA	02173-	(617) 555-5336	
Butterfield	Joyce	119 Adams Street	Cambridge	MA	02140-	(617) 555-2232	
Chao	Shih Lin	59 Holyoke Street	Lexington	MA	02173-	(617) 555-5921	
Charlton	Doug	57 Webster Street	Concord	MA	01742-	(508) 555-9246	

- 3 Select File>Save Rows As from the menu bar.
- 4 Choose the HTML Table format for the file from the Save As Type dropdown listbox.
- 5 Name the file.  
InfoMaker creates a file using the name you supplied and the extension *htm*.
- 6 Open a browser such as Netscape or Internet Explorer.
- 7 Use the browser's file open command to open the HTML file.

This screen shows the file opened in Netscape:



## Working with PSR files

A PSR file is a special file with the extension PSR created by PowerBuilder, InfoMaker, or DataWindow Builder. PSR stands for Powersoft report.

A PSR file contains a report definition (source and object) as well as the data contained in the report when the PSR file was created.

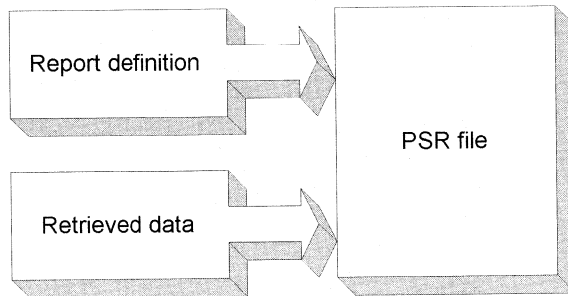
---

### About reports

A report is the same as a nonupdatable DataWindow object.

**FOR INFO** For more information, see "Reports versus DataWindow objects" on page 142.

---



You can use a PSR file to save a complete report (report design and data). This can be especially important if you need to keep a snapshot of data taken against a database that changes frequently.

#### How PSR files are created

InfoMaker creates a PSR file when you:

- ◆ Mail a report to an InfoMaker user using electronic mail.  
**FOR INFO** See "Mailing reports" on page 217.
- ◆ Save data in the Powersoft Report file format.  
**FOR INFO** See "Saving data in an external file" on page 212.

#### Opening a PSR file

When you open a PSR file, InfoMaker displays the report in the Report painter. If InfoMaker is not already running, opening a PSR file automatically starts InfoMaker.

---

**On UNIX** InfoMaker is not currently available on UNIX.

**On Macintosh** On the Macintosh, you cannot open PSR files created in versions earlier than InfoMaker 6.0.

---

InfoMaker users can open a PSR file in File Manager or Explorer, in a mail message, and using the File menu in the Report painter. PowerBuilder and DataWindow Builder users can open a PSR file in the Report painter.

**Windows and PSR files**

When InfoMaker is installed, the Powersoft report file type is registered with Windows.

❖ **To open a PSR file in InfoMaker using Explorer or File Manager or from a mail message:**

- ◆ Double-click the PSR filename.

InfoMaker displays the report in preview.

❖ **To open a PSR file from the menu bar in the Report painter in PowerBuilder, DataWindow Builder, or InfoMaker:**

- 1 Select File>Open File from the menu bar.

The Select a File Name dialog box displays.

- 2 Select the PSR file you want. Change drives and directories if needed.

The Report painter displays the report in the workspace.

- 3 Click the Preview button to preview the report.

#### PSR files and retrieval

When you are previewing a PSR file, you see the data that was saved in the file when it was created. This is true until you explicitly retrieve data again using the Retrieve button or the Rows>Retrieve command.

If you attempt to retrieve data with a PSR file, you must be sure that you are properly connected to the right database. Otherwise, you will receive a database error message.

If you retrieve new data while previewing a PSR file, you cannot go back to the old data contained in the file. To go back to the old data, you must leave the PSR file without saving and then reopen the PSR file.

## Mailing reports

On Windows systems, while previewing in the Report painter, you can mail a report as a PSR file to an InfoMaker user who is using a MAPI-compliant mail system, such as Microsoft Exchange. (MAPI stands for *messaging application program interface* and is one of the programming interfaces to mail systems.)

FOR INFO For more about PSR files, see "Working with PSR files" on page 215.

---

### **Using 16-bit mail clients**

Use of the 16-bit mail API (MAPI) is not supported under 32-bit InfoMaker—you need to use the 32-bit version.

---

#### **❖ To mail a report:**

- 1 While previewing a report in the Report painter, select File>Send from the menu bar.

If you are not logged on to your mail system, you will be prompted for your password.

- 2 Enter your password and click OK.

A form for mailing the report displays. InfoMaker creates and attaches the appropriate PSR file (which holds the report and data).

- 3 Complete the form and send the message.

InfoMaker mails the PSR file.

The recipient can open the report by double-clicking it if InfoMaker is installed.

## **Working in a grid report**

If you are previewing a grid-style report, you can do the following:

- ◆ Resize columns
- ◆ Reorder columns
- ◆ Copy data to the clipboard

#### **❖ To resize a column in a grid report:**

- 1 Position the mouse pointer at a column boundary in the header.

The pointer changes to a 2-headed arrow.

- 2 Press and hold the left mouse button (mouse button on the Macintosh) and drag the mouse to move the boundary.

- 3 Release the mouse button when the column is the correct width.

❖ **To reorder columns in a grid report:**

- 1 Press and hold the left mouse button (mouse button on the Macintosh) on a column heading.

InfoMaker selects the column and displays a line representing the column border:

Employee ID	Department	Name	Salary	Status

- 2 Drag the mouse left or right to move the column.
- 3 Release the mouse button.

❖ **To copy data to the clipboard from a grid report:**

- 1 Select the cells whose data you want to copy to the clipboard:
  - ◆ To select an entire column, click its header. To select neighboring columns, press and hold **SHIFT**, then click the headers. To select non-neighboring columns, press and hold **CTRL** (**COMMAND** on the Macintosh), then click the headers.
  - ◆ To select cells, press the left mouse button (mouse button on the Macintosh) on the bottom border of a cell and drag the mouse. Selected cells are highlighted.
- 2 Select **Edit>Copy** from the menu bar.  
*or*  
Press **CTRL+C** (**COMMAND + C** on the Macintosh).

The contents of the selected cells are copied to the clipboard. If you copied the contents of more than one column, the data is separated by tabs.

## Modifying general report properties

This section describes the general report properties that you can modify in the workspace. It covers:

- ◆ "Changing the report style" next
- ◆ "Setting colors in a report" on page 221
- ◆ "Specifying properties of a grid report" on page 222
- ◆ "Specifying pointers for a report" on page 222
- ◆ "Defining print specifications for a report" on page 223
- ◆ "Modifying text in a report" on page 227
- ◆ "Naming objects in a report" on page 228
- ◆ "Using borders in a report" on page 229
- ◆ "Specifying variable-height detail bands in a report" on page 229
- ◆ "Modifying the data source of a report" on page 232

## Changing the report style

The general style properties for a report include:

- ◆ The unit of measure used in the report
- ◆ A timer interval for events in the report
- ◆ A background color for the report

InfoMaker assigns defaults when it generates the basic report. You can change the defaults.

### ❖ To change the default style properties:

- 1 Position the pointer in the background of the report, display the popup menu, and select Properties.

The Report Object property sheet displays with the General property page on top.

- 2 Click the unit of measure you want to use to specify distances when working with the report:
  - ◆ PowerBuilder units (PBUs)
  - ◆ Pixels (smallest element on the display monitor)



- ◆ Thousandths of an inch
- ◆ Thousandths of a centimeter

---

### Choosing the unit of measure

If you plan to print the contents of the report, change the unit of measure to inches or centimeters to make it easier to specify the margin measurements.

You may also want to use inches or centimeters if you plan to move your PBL to another platform (such as from Windows to the Macintosh) because the results will be more consistent.

---

- 3 Specify the number of milliseconds you want between internal timer events in the report.

This value determines how often InfoMaker updates the time fields in the report. The default is 60,000 milliseconds (1 minute).

- 4 Select a background color from the Color dropdown listbox. The default color is the window background color.
- 5 Click OK.

## Setting colors in a report

You can set different colors for each element of a report to enhance the display of information.

### ❖ To set colors in a report:

- ◆ Do one of the following:

To set colors for	Do this
The report's background	Position the mouse on an empty spot in the report, display the popup menu, then select Properties. On the General property page of the report's property sheet, select a color from the Color dropdown listbox
A band	Position the mouse pointer on the bar that represents the band, display the popup menu, then select Properties. On the General property page of the band's property sheet, select a color from the Color dropdown listbox. The choice you make here overrides the background color for the report

To set colors for	Do this
An object	Position the mouse pointer on the object, display the popup menu, then select Properties. For objects that use text, you can set colors for background and text on the Font property page of the property sheet. For drawing objects, you can set colors on the General property page of the property sheet

## Specifying properties of a grid report

In grid reports, you can specify:

- ◆ When grid lines are displayed
- ◆ How you can interact with the report during execution

### ❖ To specify basic grid report properties:

- 1 Position the mouse pointer on the background in a grid report, display the popup menu, and select Properties.
- 2 Select the options you want in the Grid box on the General property page:

Option	Result
On	Grid lines always display
Off	Grid lines never display (you cannot resize columns during execution)
Display Only	Grid lines display only when the report displays online
Print Only	Grid lines display only when the contents of the report are printed
Column Moving	Columns can be moved during execution
Mouse Selection	Data can be selected during execution (and, for example, copied to the clipboard)
Row Resize	Rows can be resized during execution

## Specifying pointers for a report

Just as with colors, you can specify different pointers to use when the mouse is over a particular area of the report.

**❖ To change the mouse pointer used during execution:**

- 1 Position the mouse over the element of the report whose pointer you want to define, display the popup menu, and select Properties to display the appropriate property sheet.

You can set a pointer for the entire report, specific bands, and specific objects.

- 2 Select the Pointer tab.
- 3 Choose the pointer either from the Stock Pointers list or, if you have a file containing pointer definitions (CUR files), enter a pointer filename.

You can use the Browse button to search for the file.

---

**Pointers for reports running on both Windows and Macintosh**

If you design pointers for reports to be used on both platforms, use only the middle of the image. The outside will not be visible on the Macintosh. On Windows, pointers and icons are 32x32 pixels. On the Macintosh, they are 16x16 pixels.

---

- 4 Click OK.

## Defining print specifications for a report

When you are satisfied with the look of the report, you can define the print specifications for the report.

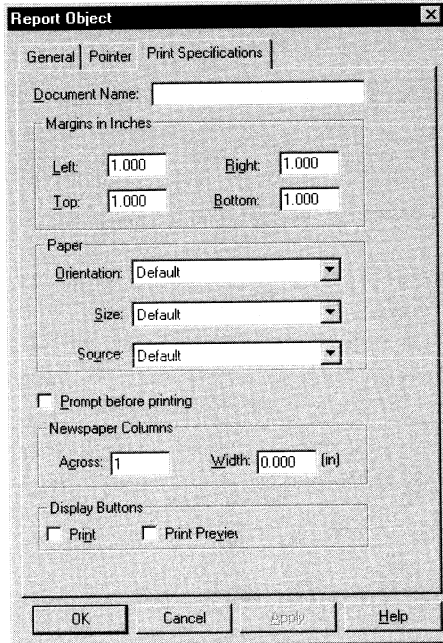
**❖ To define print specifications for a report:**

- 1 In the Report painter, select Properties from the report's popup menu to display the report's property sheet.
- 2 In the Units box on the General property page, select a unit of measure.

It is easier to specify the margins when the unit of measure is inches or centimeters.

- 3 Select the Print Specifications tab.

The Print Specifications property page uses the units of measure you specified on the General property page. For example, if the unit of measure is inches, the Print Specifications property page looks like this:



- 4 Specify a name in the Document Name box.

This is the name that will be used in the print queue to identify the report.

- 5 Specify the margins for the report.

You can also change margins in preview while you are actually looking at data. If you change margins while previewing, the changes are recorded here on the Print Specifications page.

FOR INFO See "Using preview" on page 210.

- 6 Select the paper's orientation, size, and source from the dropdown listboxes.

For orientation, choose from the following:

Setting	Result
Default	Uses the default printer setup
Portrait	Prints the contents of the report across the width of the paper

Setting	Result
Landscape	Prints the contents of the report across the length of the paper

- 7 If you want to prompt for print setup before printing, select the Prompt Before Printing checkbox.  
InfoMaker will display the standard Print Setup dialog box each time you make a print request.
- 8 If you want a multiple-column report where the data fills one column on a page, then the second, and so on, as in a newspaper, select the number and width of the columns in the Newspaper Columns box.  
FOR INFO See "Printing with newspaper-style columns" next.
- 9 Click OK.

## Printing with newspaper-style columns

When you define a report, you can specify that it print in multiple columns across the page, like a newspaper. A typical use of newspaper-style columns is a phone list, where you want to have more than one column of names on a printed page. The following report (shown in print preview) has two newspaper-style columns:

Employee Phone List				
4/23/87				
Employee Name		Phone	Employee Name	Phone
Whitney, Fran	(617) 555-3985	Chao, Shih Lin	(617) 555-5321	
Cobb, Matthew	(617) 555-3840	Blake, Barbara	(617) 555-9345	
Chen, Philip	(404) 555-2341	Smith, Susan	(713) 555-6613	
Jordan, Julie	(617) 555-7835	Preston, Mark	(617) 555-5862	
Breaull, Robert	(617) 555-3099	Clark, Alison	(510) 555-9437	
Espinosa, Melissa	(508) 555-2319	Soo, Hing	(617) 555-8748	
Bertrand, Jeannette	(508) 555-8138	Goggin, Kevin	(617) 555-3785	
Dill, Marc	(617) 555-2144	Bucceri, Matthew	(617) 555-5336	

### ❖ To define newspaper-style columns for a report:

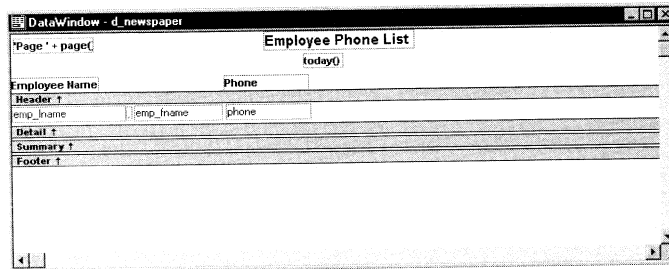
- 1 Build a tabular report with the data you want.
- 2 Select Properties from the report's popup menu.
- 3 Select the Print Specifications tab.
- 4 Specify the number of columns across the page and the width of columns in the Newspaper Columns box.
- 5 Click OK.

- 6 For each object in the report that you do *not* want to appear multiple times on the page (such as headers), select Properties from the object's popup menu. Then select the Suppress Print After First Newspaper Column checkbox.

Example

This example describes how you would create a newspaper-style report.

First create a tabular report with the last name, first name, and phone number columns. Then add a title, page number, and date. The report looks like this in the workspace:



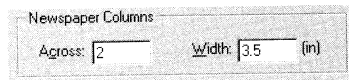
---

### Sliding columns

The Emp\_Fname column and the text object holding a comma are defined as Slide Left so they will display just to the right of the Emp\_Lname column.

---

Next you specify two columns across and a column width of 3.5 inches in the Newspaper Columns box in the Print Specifications page of the report's property sheet:



Now you're ready to preview the report.

When you preview the report, InfoMaker displays the result set in two columns. Everything above the column headers (which includes page number, title, and date) also shows twice because of the 2-column specification. This information should appear only once per page.

To specify that page number, title, and date should appear only once on the page, you need to suppress printing after the first column. For each of these objects, select Properties from the object's popup menu. Then select the Suppress Print After First Newspaper Column checkbox:

Suppress Print After First Newspaper Column

The finished report looks like this, with one set of page heading information and two columns of column header and detail information:

Employee Phone List			
4/23/96			
Employee Name	Phone	Employee Name	Phone
Whitney, Fran	(617) 555-3985	Chao, Shih Lin	(617) 555-9321
Cobb, Matthew	(617) 555-3840	Blakie, Barbara	(617) 555-9345
Chan, Philip	(404) 555-2341	Smith, Susan	(713) 555-6613
Jordan, Julie	(617) 555-7835	Preston, Mark	(617) 555-5862
Breault, Robert	(617) 555-3099	Clark, Alison	(510) 555-9437
Espinosa, Melissa	(508) 555-2319	Soo, Hing	(617) 555-8748
Bertrand, Jeanette	(508) 555-8138	Goggin, Kevin	(617) 555-3785
Dill, Marc	(617) 555-2144	Bucceri, Matthew	(617) 555-5336

## Modifying text in a report

When InfoMaker initially generates the basic report, it uses the following:

- ◆ For the text and alignment of column headings and labels, InfoMaker uses the extended column attributes made in the Database painter.
- ◆ For fonts, InfoMaker uses the definitions made in the Database painter for the table.

You can override any of these defaults in a particular report.

### ❖ To change text in a report:

- 1 Select the text.

The first box in the StyleBar is now active.

- 2 Type the new text.

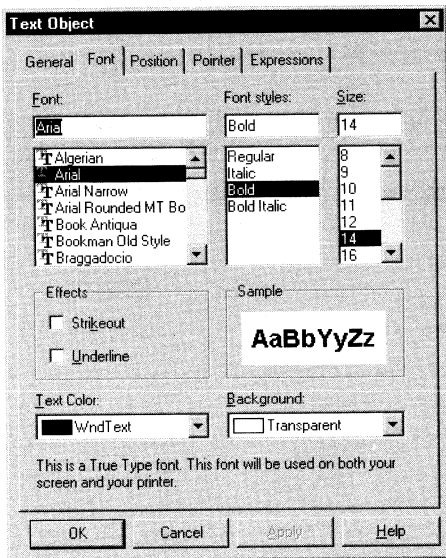
Use ~n~r to embed a newline character in the text.

### ❖ To change the text properties for a text object in a report:

- 1 Select the text object.

- 2 Do one of the following:

- ◆ Change the text properties in the StyleBar.
- ◆ Select the Font property page in the object's property sheet and change the properties there.



## Naming objects in a report

You use names to identify columns and other objects in filters and in InfoMaker functions. InfoMaker names objects it places in the generated report (that is, columns, labels, and headings), but you need to name objects you place yourself if you want to refer to them anywhere.

- ❖ **To specify a name for an object in a report:**
  - 1 Select Properties from the object's popup menu and then select the General tab.
  - 2 Type the name in the Name box.



## Using borders in a report

You can place borders around text, columns, graphs, and crosstabs to enhance their appearance. InfoMaker provides six types of borders: Underline, Box, Resize, Shadow box, 3D Raised, and 3D Lowered:

Amo & Sons    Laura    **McCarthy**    1210 Highway 36    Carmel    IN

If you specify the Resize border, you can resize the object during execution. Resize borders are particularly useful with graphs.

---

### Border appearance varies

Changing the border style may not have the same effect on all platforms. For example, on Windows 95, the 3D Lowered style looks the same as the Box style for DropDownListBoxes.

---

#### ❖ To add a border to an object in a report:

- 1 Select one or more objects.
- 2 Select the border you want from the Border dropdown toolbar in the PainterBar.

InfoMaker places the border around the selected objects.

You can also specify a border for an object in the object's property sheet on the General property page. But when you use the property sheet, you can deal with only one object at a time. When you use the Border dropdown toolbar, you can change borders for many objects at once.

## Specifying variable-height detail bands in a report

Sometimes reports contain columns whose data is of variable length. For example, a Memo column in a table might be a character column that can take up to several thousand characters. You don't want to reserve space for that much information for the column in the detail band, since it would make the detail band's height very large, meaning you could see few rows at a time.

Instead, you want the detail band to resize based on the data in the Memo column. If the Memo column has only one line of text, you want the detail band to be one line. If the Memo column has 20 lines of text, you want the detail band to be 20 lines high.

To provide a detail band that resizes as needed, you specify that the variable-length columns and the band have Autosize Height.

❖ **To create a resizable detail band in a report:**

- 1 Select Properties from the popup menu of a column that should resize based on the amount of data.
- 2 Select the Autosize Height checkbox on the Position property page and click Apply.
- 3 Clear the Auto Horz Scroll checkbox on the Edit property page and click OK.

InfoMaker will wrap text during preview instead of displaying text on one scrollable line.

- 4 Repeat steps 1 to 3 for any other columns that should resize.
- 5 Select Properties from the detail band's popup menu.
- 6 Select the Autosize Height checkbox and click OK.

During preview, the detail band will resize based on the contents of the columns you defined as having Autosize Height.

---

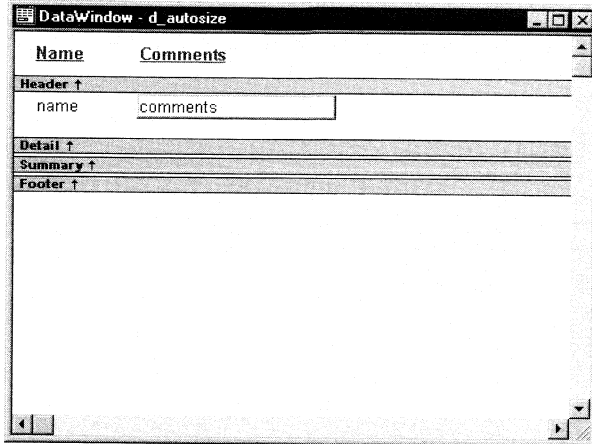
**Clipping columns**

You can have Autosize Height columns without an Autosize Height detail band. If such a column expands beyond the size of the detail band during preview, it will be clipped.

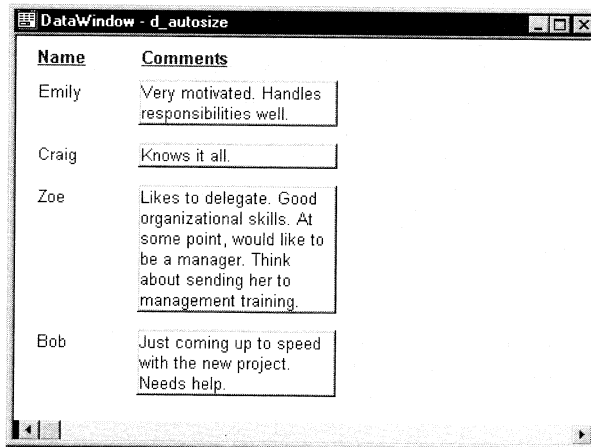
---

## Example

In the following report, the Comments column is defined as having Autosize Height, as is the detail band. Horizontal scrolling is turned off for the Comments column:



During execution, the space allocated for a row depends on how much information is in Comments; the detail band expands to fit the text. In the report below, the first row takes two lines, the second row takes one line, and so on:



## Modifying the data source of a report

When modifying a report, you might realize that you haven't included all the columns you need. Or you might need to define retrieval arguments. You can modify the data source from the Report painter workspace. How you do it depends on the data source.

## Modifying SQL SELECT statements

If the data source is SQL (such as Quick Select, SQL Select, or Query), you can graphically modify the SQL SELECT statement.

### ❖ To modify a SQL data source:

- 1 Select Design>Edit Data Source from the menu bar.

InfoMaker returns you to the Select painter. (If you used Quick Select to define the data source, this might be the first time you have seen the Select painter.)

- 2 Modify the SELECT statement graphically using the same techniques as when creating it.

**FOR INFO** For more information, see "Using SQL Select" on page 166.

---

### Modifying the statement syntactically

Select Design>Convert to Syntax from the menu bar to modify the SELECT statement syntactically.

---

- 3 Select Design>Data Source from the menu bar to return to the painter workspace.

---

### Changing the table

If you change the table referenced in the SELECT statement, InfoMaker maintains the columns in the workspace (now from a different table) only if they match the data types and order of the columns in the original table.

---

Modifying the  
retrieval arguments

You can add, modify, or delete retrieval arguments when modifying your data source.

**❖ To modify the retrieval arguments:**

- 1 In the Select painter, select Design>Retrieval Arguments from the menu bar.

The Specify Retrieval Arguments dialog box displays listing the existing arguments.

- 2 Add, modify, or delete the arguments.
- 3 Click OK.

You return to the Select painter, or to the text window displaying the SELECT statement if you are modifying the SQL syntactically.

- 4 Reference any new arguments in the WHERE or HAVING clause of the SELECT statement.

**FOR INFO** For more information about retrieval arguments, see Chapter 4, "Defining Reports".

**Modifying the result set**

If the data source is External or Stored Procedure, you can modify the result set description.

**❖ To modify a result set:**

- 1 Select Design>Edit Data Source from the menu bar.

The Modify Result Set Description dialog box displays.

- 2 Review the description and make any necessary changes.
- 3 Click OK.

If the data source is a stored procedure

If you are modifying the result set for a report whose data source is a stored procedure, the Modify Result Set Description dialog box contains a More button.

Click More to edit the Execute statement, select another stored procedure, or add retrieval arguments:

Name	Type	Length	Dec
d	long		
company_name	string	35	

OK  
Cancel  
Add  
Delete  
Insert  
Help

Edit Execute: Result Set: 1

execute dba.sp\_customer\_list,0 Procedures...  
Arguments...

Paste Arguments: [ ]

## Reorganizing objects in a report

This section describes the activities that help you change the layout and appearance of the objects in a report, including:

- ◆ "Displaying boundaries for objects in a report" next
- ◆ "Using the grid and the ruler in a report" on page 236
- ◆ "Deleting objects in a report" on page 236
- ◆ "Moving objects in a report" on page 237
- ◆ "Copying objects in a report" on page 237
- ◆ "Resizing objects in a report" on page 238
- ◆ "Aligning objects in a report" on page 238
- ◆ "Equalizing the space between objects in a report" on page 239
- ◆ "Equalizing the size of objects in a report" on page 239
- ◆ "Sliding objects to remove blank space in a report" on page 240

## Displaying boundaries for objects in a report

When reorganizing objects in the workspace, it is sometimes helpful to see how large all the objects are. That way, you can easily check for overlapping objects and make sure that the spacing around objects is what you want.

### ❖ To display object boundaries in a report:

- 1 Select Design>Options from the menu bar.  
The Report Options property sheet displays.
- 2 Select the Show Edges checkbox on the General property page.  
InfoMaker displays the boundaries of each object in the report.

---

### **Boundaries display only in the painter workspace**

The boundaries displayed for objects are for use only in the painter workspace. They do not display in a running report or in a printed report.

---

## Using the grid and the ruler in a report

The Report painter provides a grid and a ruler to help you align objects.

❖ **To use the grid and the ruler:**

- 1 Select Design>Options from the menu bar.

The Report Options property sheet displays. The Alignment Grid box on the General property page contains the alignment grid options.

- 2 Use the options as needed:

Option	Meaning
Snap to Grid	Make objects snap to a grid position when you place them or move them
Show Grid	Show or hide the grid when the workspace displays
X	Specify the size (width) of the grid cells
Y	Specify the size (height) of the grid cells
Show Ruler	Show a ruler. The ruler uses the units of measurement specified in the Style dialog box FOR INFO See "Changing the report style" on page 220

Your choices for the grid and the ruler are saved and used the next time you start InfoMaker.

## Deleting objects in a report

❖ **To delete objects in a report:**

- 1 Select the objects you want to delete.
- 2 Click the Clear button.  
*or*  
Select Edit>Clear from the menu bar.  
*or*  
Press the DEL key.



## Moving objects in a report

In all presentation styles except Grid

In all presentation styles except Grid, you can move all the objects (such as headings, labels, columns, drawing objects, and graphs) anywhere you want.

❖ **To move objects in a report:**

- 1 Select the objects you want to move.
- 2 Do one of the following:
  - ◆ Drag the objects with the mouse.
  - ◆ Press an arrow key to move the objects in one direction.

In grid reports

You can reorder columns in a grid report when previewing.

FOR INFO See "Working in a grid report" on page 218.

## Copying objects in a report

You can copy objects within a report and to other reports. All properties of the objects are copied.

❖ **To copy an object in a report:**

- 1 Select the object in the Report painter workspace.
- 2 Select Edit>Copy from the menu bar.  
*or*  
Press CTRL+C (COMMAND+C on the Macintosh).  
The object is copied to a private InfoMaker clipboard.
- 3 Copy (paste) the object to the same report or to another one:
  - ◆ To copy the object within the same report, select Edit>Paste from the menu bar or press CTRL+V (COMMAND +V on the Macintosh).
  - ◆ To copy the object to another report, open another instance of the Report painter and open the desired report in it. Make that report active and paste the object.

InfoMaker pastes the object at the same location as in the source report (if you are pasting into the same report, you should move the pasted object so it doesn't cover the original object). InfoMaker displays a message box if the object you are pasting is not valid in the destination report.

## Resizing objects in a report

You can resize an object using the mouse or the keyboard.

### Using the mouse

To resize an object using the mouse, select it, then grab an edge and drag it with the mouse.

### Using the keyboard

To resize an object using the keyboard, select it, then do the following:

To make the object	Press
Wider	SHIFT+RIGHT ARROW
Narrower	SHIFT+LEFT ARROW
Taller	SHIFT+DOWN ARROW
Shorter	SHIFT+UP ARROW

### In grid reports

You can resize columns in grid reports.

#### ❖ To resize a column in a grid report:

- 1 Position the mouse pointer at a column boundary.  
The pointer changes to a 2-headed arrow.
- 2 Press and hold the left mouse button (mouse button on the Macintosh) and drag the mouse to move the boundary.
- 3 Release the mouse button when the column is the correct width.

## Aligning objects in a report

Often you want to align several objects or make them all the same size. You can use the grid to align the objects or you can have InfoMaker align them for you.

#### ❖ To align objects in a report:

- 1 Select the object whose position you want to use to align the others.  
InfoMaker displays handles around the selected object.
- 2 Extend the selection by pressing and holding the CTRL key (COMMAND on the Macintosh) and clicking the objects you want to align with the first one.

All the objects have handles on them.

- 3 Select Edit>Align Objects from the menu bar.
- 4 From the cascading menu, select the dimension along which you want to align the controls.

For example, to align the objects along the left side, select the first choice on the cascading menu.

InfoMaker moves all the selected objects to align with the first one.

## Equalizing the space between objects in a report

If you have a series of objects and the spacing is fine between two of them but the spacing is wrong for the rest, you can easily equalize the spacing around all the objects.

### ❖ To equalize the space between objects in a report:

- 1 Select the two objects whose spacing is correct (click one object, then press CTRL (COMMAND on the Macintosh) and click the second object).
- 2 Select the other objects whose spacing you want to be the same as the first two objects by pressing CTRL (COMMAND on the Macintosh) and clicking.
- 3 Select Edit>Space Objects from the menu bar.
- 4 From the cascading menu, select the dimension whose spacing you want to equalize.

## Equalizing the size of objects in a report

Say you have several objects in a report and want their sizes to be the same. You can accomplish this manually or by using the Edit menu.

### ❖ To equalize the size of objects in a report:

- 1 Select the object whose size is correct.
- 2 Select the other objects whose size you want to match the first object by pressing CTRL (COMMAND on the Macintosh) and clicking.
- 3 Select Edit>Size Objects from the menu bar.
- 4 From the cascading menu, select the dimension whose size you want to equalize.

## Sliding objects to remove blank space in a report

You can specify that you want to eliminate blank lines or spaces in a report by sliding columns and other objects in the report to the left or up if there is blank space. You can use this feature to remove blank lines in mailing labels or to remove extra spaces between fields (such as first name and last name).

---

### **Slide is used by default in nested reports**

InfoMaker uses slide options automatically when you nest a report to ensure that the reports are positioned properly.

---

#### ❖ **To use sliding columns or objects in a report:**

- 1 Select Properties from the object's popup menu and then select the Position tab.
- 2 Select the Slide options you want:

<b>Option</b>	<b>Description</b>
Left	Slide the column or object to the left if there is nothing to the left. Be sure the object does not overlap the object to the left. Sliding left will not work if the objects overlap
Up - All Above	Slide the column or object up if there is nothing in the row above (the row above must be completely empty for the column or object to slide up)
Up - Directly Above	Slide the column or object up if there is nothing <i>directly above it</i> in the row above

---

### **If you are sliding columns up**

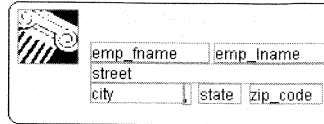
Even blank columns have height; if you want columns to slide up, you need to specify as Autosize Height all columns above that might be blank and that you want to slide other columns up through.

---

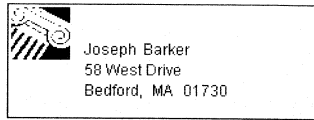
#### Example

In a mailing label that includes first and last names, as well as address information, you can use sliding to combine the columns appropriately.

In the following label, `emp_fname`, the comma, `state`, and `zip_code` are specified as slide left. Edges are shown to indicate the spacing between the columns. Notice that there is a small amount of space between objects. This space is necessary for Slide Left to work properly:



When you preview (run) the report, the last name, comma, `state`, and `zip code` slide left to remove the blank space:



## Prompting for retrieval criteria in a report

You can define your report so that it always prompts for retrieval criteria just before it retrieves data.

❖ **To prompt for retrieval criteria in a report:**

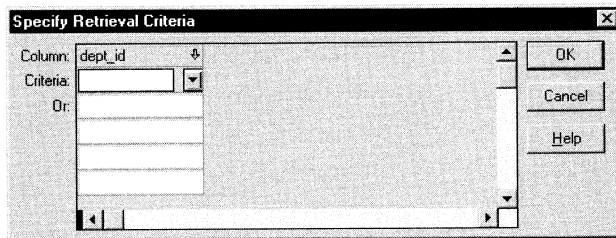
- 1 Select Rows>Prompt for Criteria from the menu bar.

The Prompt For Criteria dialog box displays listing all columns in the report.

- 2 Select the columns you want to be able to specify retrieval criteria for when you run the report and click OK.

What happens

When you specify prompting for criteria, InfoMaker displays the following dialog box just before a retrieval is to be done:



Each column you selected in the Prompt for Criteria dialog box displays in the grid. You can specify criteria here exactly as in the grid with the Quick Select data source.

Criteria specified here are added to the WHERE clause for the SQL SELECT statement defined for the report.

---

### Testing in InfoMaker

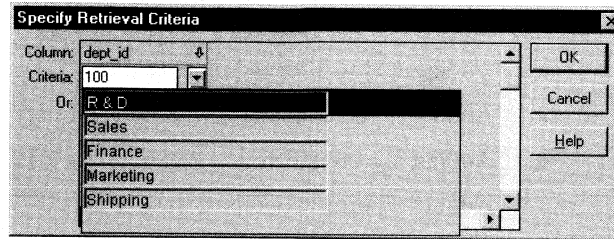
You can test the prompting for criteria by previewing the report. Note that by default, after doing the first retrieval, InfoMaker stores the data internally (caches it) and doesn't re-retrieve it the next time you go to preview. So you won't be prompted for criteria each time you go to preview. You can explicitly retrieve by clicking the Retrieve button.

FOR INFO For more about data caching, see "Retrieving data" on page 206.

---

**Using edit styles**

If a column uses a code table or the RadioButton, CheckBox, or DropDownListBox edit style, an arrow displays in the column header and you can select a value from a dropdown listbox when specifying criteria:



If you don't want the dropdown listbox used for a column for specifying retrieval criteria, select the Override Edit checkbox on the General property page of the column's property sheet.

**Forcing the entry of criteria**

If you have specified prompting for criteria for a column, you can force the entry of criteria for the column by selecting the Equality Required checkbox on the General property page of the column's property sheet. InfoMaker will underline the column header in the grid during prompting. Selection criteria for the specified column must be entered, and the = operator must be used.

**For more information**

The section "Using Quick Select" on page 156 describes in detail how you can specify selection criteria in the grid.

## Adding objects to a report

This section describes adding objects to enhance your report:

- ◆ "Adding columns to a report" next
- ◆ "Adding text to a report" on page 245
- ◆ "Adding drawing objects to a report" on page 245
- ◆ "Adding a groupbox to a report" on page 246
- ◆ "Adding pictures to a report" on page 246
- ◆ "Adding computed fields to a report" on page 248
- ◆ "Adding buttons to a report" on page 253
- ◆ "Adding graphs to a report" on page 255
- ◆ "Adding OLE objects to a report" on page 255
- ◆ "Adding reports to a report" on page 255

## Adding columns to a report

You can add columns that are included in the data source to a report. When you first create a report, each of the columns in the data source is automatically placed in the report. Typically, you would add a column to restore one that you had deleted from the report, or to display the column more than once in the report.

---

### **Adding columns not previously retrieved to the data source**

To specify that you want to retrieve a column not previously retrieved (that is, add a column to the data source), you must modify the data source.

FOR INFO See "Modifying the data source of a report" on page 232.

---

#### ❖ **To add a column from the data source to a report:**

- 1 Click the Column button.  
*or*  
Select Objects>Column from the menu bar.

- 2 Click where you want to place the column.

The Select Column dialog box displays listing all columns included in the data source of the report.



- 3 Select the column and click OK.

## Adding text to a report

When InfoMaker generates a basic report from a presentation style and data source, it places columns and their headings in the workspace. You can add text anywhere you want to make the report easier to understand.

### ❖ To add text to a report:

- 1 Click the Text button in the Objects dropdown toolbar.  
*or*  
Select Objects>Text from the menu bar.
- 2 Click where you want the text.  
  
InfoMaker places the text object in the workspace and displays the word *text*.
- 3 Type the text in the textbox at the top of the workspace.
- 4 (Optional) Change the font, size, style, and alignment for the text using the StyleBar.

## Adding drawing objects to a report

You can add the following drawing objects to a report to enhance its appearance:

Rectangle  
RoundRectangle  
Line  
Oval

### ❖ To place a drawing object in a report:

- 1 Select the drawing object from the Objects dropdown toolbar or from the Objects menu.
- 2 Click where you want the object to display.
- 3 Resize or move the drawing object as needed.

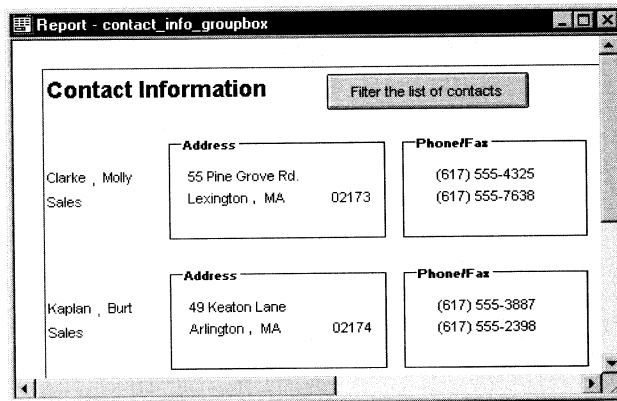
- 4 Use the drawing object's property sheet to change its properties as needed.

For example, you might want to specify a fill color for a rectangle or thickness for a line.

## Adding a groupbox to a report

To visually enhance the layout of a report, you can add a groupbox. The groupbox is a static frame used to group and label a set of objects in a report.

The following example shows two groupboxes in a report. The Address groupbox groups address information and the Phone/Fax groupbox groups telephone numbers:



### ❖ To add a groupbox to a report:

- 1 Select Objects>Group Box from the menu bar and click in the workspace.
- 2 With the groupbox selected, type the text to display in the frame.
- 3 Move and resize the groupbox as desired.

## Adding pictures to a report

You can place pictures, such as your company logo, in a report to enhance its appearance. If you place a picture in the header, summary, or footer band of the report, the picture displays each time the contents of that band displays. If you place the picture in the detail band of the report, it displays in each row.

**Tips for using pictures**

To display a different picture for each row of data, retrieve a column containing picture filenames from the database.

**FOR INFO** For more information, see "Modifying column properties in the Database painter" on page 84.

If you change the bitmap in the Picture control in a report, you need to reset the original size property. The property automatically reverts to the default setting when you change the bitmap.

---

**❖ To place a picture in a report:**

- 1 Click the Picture button in the Objects dropdown toolbar.  
*or*  
Select Objects>Picture from the menu bar.
  - 2 Click where you want the picture to display.  
The Picture Object property sheet displays.
  - 3 Use the Browse button to find the file or enter a filename in the File Name box.  
  
On Windows, the picture must be a bitmap (BMP), runlength-encoded (RLE), or Windows metafile (WMF) file.  
  
On Macintosh systems, the picture must be a bitmap (BMP), runlength-encoded (RLE), or PICT file.
- 

**Naming PICT files**

If you are using a PICT file and the report will be bundled into an executable file, the PICT file must be named in the format NAME.PCT. Otherwise, when you run the report from the executable, the image will not display.

---

- 4 Name the picture by typing a name in the Name box.
- 5 Click the Original Size checkbox to display the bitmap in its original size.  
  
You can use the mouse to change the size of the bitmap in the Report painter.
- 6 Click the Invert Image checkbox to display the picture with its colors inverted.

- 7 Click the OK button.

You return to the workspace with the picture in place.

## Adding computed fields to a report

You can use **computed fields** in any band of the report. Typical uses include:

- ◆ **Calculations based on column data that change for each retrieved row** For example, if you are retrieving yearly salary, you can define a computed field in the detail band that displays monthly salary (defined as `Salary / 12`).
- ◆ **Summary statistics of the data** For example, if you have a grouped report, you can use a computed field to calculate the totals of a column for each group.
- ◆ **Concatenated fields** For example, if you are retrieving first name and last name, you can define a computed field that concatenates the values so they appear with only one space between them (defined as `Fname + " " + Lname`).
- ◆ **System information** For example, you can place the current date and time in a report's header by using computed fields (defined as `Today()` and `Now()`).

## Computed columns versus computed fields

When creating a report, you can define computed columns and computed fields as follows:

- ◆ In the Select painter, you can define **computed columns** when you are defining the SELECT statement that will be used to retrieve data into the report
- ◆ In the Report painter workspace, you can define **computed fields** after you have defined the SELECT statement (or other data source)

The difference  
between the two ways

When you define the computed column in the Select painter, the value is calculated by the DBMS when the data is retrieved. The computed column's value does not change until data has been updated and retrieved again.

When you define the computed field in the Report painter workspace, the value of the column is calculated in the report after the data has been retrieved.

## Defining a computed field in the Report painter workspace

### ❖ To define a computed field in the Report painter workspace:

- 1 Click the Compute button in the Objects dropdown toolbar.  
*or*  
Select Objects>Computed Field from the menu bar.
- 2 Click where you want the computed field.

If the calculation is to be based on column data that changes for each row, make sure you place the computed field in the detail band.

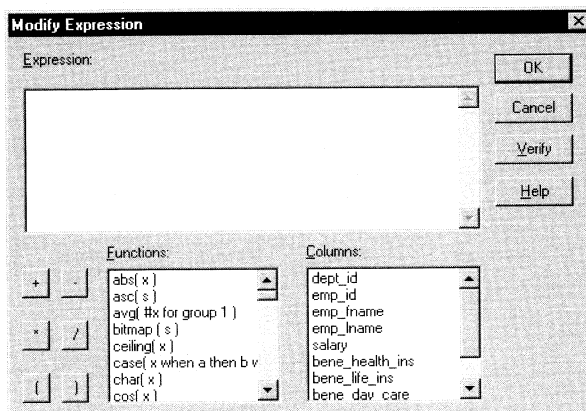
The Computed Object property sheet displays:

- 3 Name the computed field.
- 4 Click the More button.

The Modify Expression dialog box displays listing:

- ◆ InfoMaker functions you can use in the computed field
- ◆ The columns in the report

◆ Operators and parentheses



- 5 Enter the expression that defines the computed field (see below).
- 6 (Optional) Click Verify to test the expression.  
InfoMaker analyzes the expression.
- 7 Click OK.  
InfoMaker returns you to the computed column's property sheet.
- 8 Click OK.  
You return to the workspace with the computed field in place.

Entering the expression

You can enter any valid InfoMaker expression when defining a computed field. You can paste operators, columns, and InfoMaker functions into the expression from information in the Modify Expression dialog box.

**InfoMaker expressions**

The expression you are entering is an InfoMaker expression; it is not a SQL expression processed by the DBMS. So the expression follows the rules for InfoMaker expressions.

FOR INFO For complete information about InfoMaker expressions, see Chapter 22, "Operators and Expressions".

You can use the + operator to concatenate strings.

Referring to next and previous rows

You can refer to other rows in a computed field. This is particularly useful in n-up reports when you want to refer to another row in the detail band. Use this syntax:

*ColumnName[x]*

where *x* is an integer. 0 refers to the current row (or first row in the detail band), 1 refers to the next row, -1 refers to the previous row, and so on.

**Examples**

Here are some examples of computed fields and columns:

<b>To display</b>	<b>Enter this expression</b>	<b>In this band</b>
Current date at top of each page	Today( ) (a built-in InfoMaker function)	Header
Current time at top of each page	Now( )	Header
Current page at bottom of each page	Page( )	Footer
Total page count at bottom of each page	PageCount( )	Footer
Concatenation of Fname and Lname columns for each row	Fname + " " + Lname	Detail
Monthly salary if Salary column contains annual salary	Salary / 12	Detail
Four asterisks if the value of the Salary column is greater than \$50,000	IF(Salary > 50000, "****", "")	Detail
Average salary of all retrieved rows	Avg(Salary)	Summary
Count of retrieved rows, assuming each row contains a value for EmpID	Count(EmpID)	Summary

**For more information**

For complete information about the functions you can use in computed fields in the Report painter, see Chapter 23, "DataWindow Painter and InfoMaker Functions".

**A shortcut for doing summary statistics**

InfoMaker provides a quick way to create computed fields that summarize values in the detail band.

❖ **To summarize values:**

- 1 Select one or more columns in the report's detail band.

- 2 Do one of the following:

To place this computed field	Do this
Average	Select Objects>Average - Computed Field from the menu bar
Count	Select Objects>Count - Computed Field from the menu bar
Sum	Select Objects>Sum - Computed Field from the menu bar or add the built-in Sum button to the PainterBar and use it

InfoMaker places a computed field in the summary band or in the group trailer band if the report is grouped. The band is resized automatically to hold the computed field. If there is already a computed field that matches the one being generated, it is skipped.

**Adding custom buttons that place computed fields** You can add buttons to the PainterBar in the Report painter that place computed fields using any of the aggregate functions, such as Max, Min, and Median.

❖ **To customize the PainterBar with custom buttons for placing computed fields:**

- 1 Place the mouse pointer over the PainterBar and select Customize from the popup menu.

The Customize dialog box displays.

- 2 Click Custom in the Select palette group to display the set of custom buttons.

- 3 Drag a custom button into the Current toolbar group and release it.

The Toolbar Item Command dialog box displays.

- 4 Click the Function button.

The Function For Toolbar dialog box displays.

- 5 Select a function and click OK.

You return to the Toolbar Item Command dialog box.

- 6 Specify text that displays for the button and click OK.

InfoMaker places the new button in the PainterBar. You can click it to add a computed field to your report the same way you use the built-in Sum button.



A shortcut for placing page numbers and date

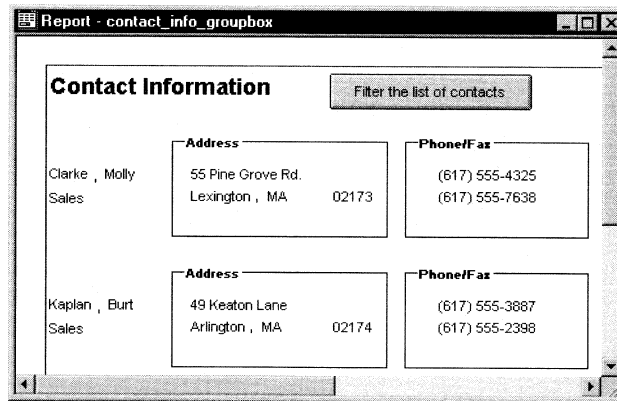
You can click buttons in the PainterBar to place a computed field for the current page number and date anywhere in the report:

To place this computed field	Do this
'Page ' + page() + ' of ' + pageCount()	Click the Page button or select Objects>Page n of n Computed Field from the menu bar
today()	Click the Today button or select Objects>Today() - Computed Field from the menu bar

## Adding buttons to a report

The Button object is a command (or Picture) style button that can be placed in a report. When clicked at execution time, the button activates the action you assign to it.

The following example shows a button placed in a report. Clicking the button brings up the Filter dialog, where you can specify a filter to be applied to the currently retrieved data:



### ❖ To add a button to a report:

- 1 Select Objects>Button from the menu bar.
- 2 Click where you want the button to display.
- 3 With the button still selected, type the text to display on the button.
- 4 Display the button's property sheet (General page).

- 5 Select the action you want to assign to the button from the Action dropdown listbox.  
  
FOR INFO For information about actions, see "Actions assignable to buttons in reports" on page 254.
- 6 If you want to add a picture to the button, select the Use Action Default Picture checkbox or enter the name of the image file to display on the button.

## Controlling the display of buttons in preview and on printed output

You can choose whether to display buttons in print preview or in printed output. You control this in the property sheet for the report (not the property sheet for the button).

- ❖ **To control the display of buttons in a report in preview and on printed output:**
  - 1 Display the report's property sheet with the Print page on top.
  - 2 Select the Print checkbox to have buttons included in the printed output when the report is printed.
  - 3 Select the Print Preview checkbox to have the buttons display on the screen when viewing the report in preview.

## Actions assignable to buttons in reports

These are the actions you can assign to a button in a report:

Action	What it does
Cancel	Cancels a retrieval that has been started with the option to yield
Filter	Displays Filter dialog and filters as specified
Page First	Scrolls to the first page
Page Last	Scrolls to the last page
Page Next	Scrolls to the next page
Page Prior	Scrolls to the prior page
Preview With Rulers	Toggles between rulers on and off
Print	Prints one copy of the DataWindow object

Action	What it does
Retrieve	Retrieves rows from the database. The option to yield is not automatically turned on
Retrieve (Yield)	Retrieves rows from the database. Before retrieval actually occurs, option to yield is turned on; this will allow the Cancel action to take effect during a long retrieve
Save Rows As	Displays Save As dialog and saves rows in the format specified
Sort	Displays Sort dialog and sorts as specified

## Adding graphs to a report

Graphs are one of the best ways to present information. For example, if your report displays sales information over the course of a year, you can easily build a graph in a report to display the information visually.

InfoMaker offers many types of graphs and provides you with the ability to control the appearance of a graph to best meet your application's needs.

**FOR INFO** For information on using graphs, see Chapter 10, "Working with Graphs".

## Adding OLE objects to a report

On Windows, you can add the following to a report:

- ◆ A column that contains a database binary large object (a blob object) using OLE 2.0
- ◆ OLE 2.0 objects

**FOR INFO** For information on using OLE in a report, see Chapter 13, "Using OLE in a Report".

## Adding reports to a report

You can nest reports in a report.

**FOR INFO** For information on nesting reports, see Chapter 9, "Using Nested Reports".

## Positioning objects in a report

Each object has several properties that determine how it is positioned within the report:

Property	Meaning
Background	Object is behind other objects. It is not restricted to one band. This is useful for adding a watermark (such as the word CONFIDENTIAL) to the background of a report
Band	Object is placed within one band. It cannot extend beyond the band's border
Foreground	Object is in front of other objects. It is not restricted to one band
Moveable	Object can be moved in preview. This is useful for designing layout
Resizable	Object can be resized in preview. This is useful for designing layout
Suppress Print After First Newspaper Column	Object appears only in the first column on the page; in subsequent columns the object does not appear. This is only for newspaper columns, where the entire report snakes from column to column

Default positioning

InfoMaker uses the following defaults when placing new objects:

Object	Default positioning
Graph	Foreground, movable, resizable
All other objects	Band, not movable, not resizable

❖ **To change the position of an object in a report:**

- 1 Select Properties from the object's popup menu and then select the Position tab.
- 2 From the Layer option dropdown listbox, select Background, Band, or Foreground.
- 3 Select Resizable or Moveable as appropriate.

## Storing data in a report

Usually you retrieve data into a report in preview—the data is changeable and you want the latest.

But sometimes the data you display in a report is static—it never changes. Or you may want a snapshot of the data at a certain point in time. In these situations, you may want to store the data in the report itself. That way you don't need to go out to the database or other data source to display the data.

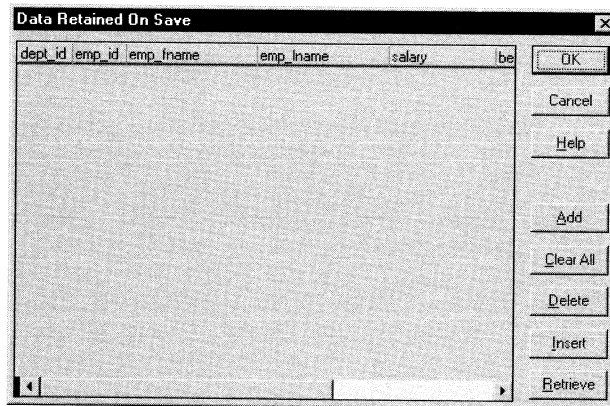
The most common reason to store data in a report is for use as a dropdown DataWindow where the data is not coming from a database. For example, you might want to display a list of state postal codes for entering values in a State column in a form. You can store those codes in a report and use the dropdown DataWindow edit style for the State column.

**FOR INFO** For more information about using the dropdown DataWindow edit style, see Chapter 6, "Displaying and Validating Data".

### ❖ To store data in a report:

- 1 Select Rows>Data from the menu bar.

The Data Retained On Save dialog box displays. All columns defined for the report are listed at the top:



- 2 Do one of the following:
  - ◆ Click Add to create an empty row and type a row of data into the window. You can enter as many rows as you want.
  - ◆ Click Retrieve to retrieve all the rows of data from the database. If you want, you can delete rows you don't want to save or manually add new rows.

---

**Data changes are local to the report**

Adding or deleting data here does not change the data in the database. It only determines what data will be stored with the report when you save it.

---

3 Click OK.

When you save the report, the data is stored in the report.

## What happens when you preview the report

When you preview a report with stored data, the data is already there. InfoMaker does not retrieve data.

With one exception, if you *do* issue Retrieve for a report stored with data, InfoMaker handles it the same as a report that is not stored with data: InfoMaker gets the latest data by retrieving rows from the database.

The exception

InfoMaker never retrieves data into a dropdown DataWindow that already contains data.

## Retrieving rows as needed

If your report retrieves hundreds of rows, there can be a noticeable delay while all the rows are retrieved. In these reports, you can have InfoMaker retrieve only as many rows as it has to before displaying data.

For example, if your report displays only 10 rows at a time, it might make sense to have InfoMaker retrieve only a small number of rows before presenting the data. Then as you page through the data, InfoMaker continues to retrieve what is necessary to display the new information. There may be slight pauses while InfoMaker retrieves the additional rows, but the pauses may be worth it if you do not need to wait a long time to start working with data.

❖ **To specify that a report retrieve only as many rows as it needs to:**

- ◆ Select Rows>Retrieve>Rows As Needed from the menu bar.

With this setting, InfoMaker presents data and returns control to you when it has retrieved enough rows to display in the report.

Retrieve Rows As Needed is overridden if you have specified sorting or have used aggregate functions, such as Avg and Sum, in the report. This is because InfoMaker must retrieve every row before it can sort or perform aggregates.

In a multiuser situation, Retrieve Rows As Needed might lock other people from the tables.

## **Saving retrieved rows to disk**

---

### **Platform note**

This feature is available on Windows 95 and Windows NT only.

---

If you want to maximize the amount of memory available to InfoMaker and other running applications, you can have InfoMaker save retrieved data on your hard disk in a temporary file rather than keep the data in memory. InfoMaker swaps rows of data from the temporary file into memory as needed to display data.

❖ **To maximize available memory by saving retrieved rows to disk:**

- ◆ Select Rows>Retrieve>Rows to Disk from the menu bar.

With this setting, when displaying data, InfoMaker swaps rows of data from the temporary file into memory instead of keeping all the retrieved rows of data in memory.



# Displaying and Validating Data

## About this chapter

This chapter describes how to customize your report or form by modifying the display values in columns and specifying validation rules.

## Contents

<b>Topic</b>	<b>Page</b>
About displaying and validating data	262
Working with display formats	264
Working with edit styles	276
Working with validation rules	292
Maintaining the entities	300

## About displaying and validating data

When InfoMaker generates a basic report or form, it uses the extended attributes defined for the data and stored in the Powersoft repository.

**FOR INFO** For more information about the Powersoft repository, see the Appendix, "The Powersoft Repository".

In the Database painter, you can create the extended attribute definitions that specify a column's display format, edit style, and validation rules.

In the Report painter or Form painter, you can override these extended attribute definitions for a column in a report or form. These overrides do not change the information stored with the column definition in the repository.

---

### **Database painter is required**

You must have the Database painter installed to define display formats and edit styles in your database.

---

## Presenting the data

When you generate a new report or form, InfoMaker presents the data according to the properties already defined for a column, such as a column's display format and edit style.

### Display formats

Display formats embellish data values while still displaying them as letters, numbers, and special characters. Using display formats, for example, you can:

- ◆ Change the color of numbers to display a negative value
- ◆ Add parentheses and dashes to format a telephone number
- ◆ Add a dollar sign and period to indicate a currency format

**FOR INFO** For information, see "Working with display formats" on page 264.

### Edit styles

Edit styles take precedence over display formats and specify how column data is presented. For example, using edit styles, you can:

- ◆ Display valid values in a dropdown listbox
- ◆ Indicate that a single value is selected by a checkbox
- ◆ Indicate which of a group of values is selected with radio buttons

In the Database painter and in the Form painter, an edit style enables you to *change* data in the database by means of an edit mechanism, such as a checkbox or radio buttons. In the Report painter, an edit style is simply a way of *presenting* data. You can change data in a form; you cannot change data in a report.

FOR INFO For more information, see "Working with edit styles" on page 276.

## **Validating data**

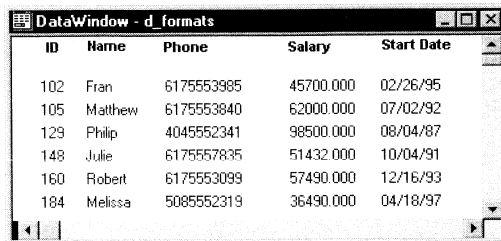
When data is entered in the Data Manipulation painter or in a form, InfoMaker evaluates the data against validation rules defined for that column. If the data is valid, InfoMaker accepts the entry; otherwise, InfoMaker displays an error message and does not accept the entry.

FOR INFO For more information, see "Working with validation rules" on page 292.

## Working with display formats

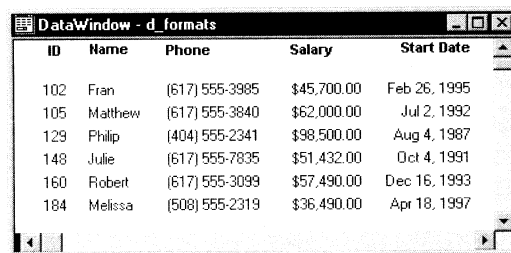
You can use display formats to customize the display of column data in a report or form. Display formats are masks in which certain characters have special significance. For example, you can display currency values preceded by a dollar sign, show dates with month names spelled out, and use a special color for negative numbers. InfoMaker comes with many predefined display formats. You can use them as is or define your own.

Here is a report without any display formats; all values display as they are stored in the database:



ID	Name	Phone	Salary	Start Date
102	Fran	6175553985	45700.000	02/26/95
105	Matthew	6175553840	62000.000	07/02/92
129	Philip	4045552341	98500.000	08/04/87
148	Julie	6175557835	51432.000	10/04/91
160	Robert	6175553099	57490.000	12/16/93
184	Melissa	5085552319	36490.000	04/18/97

Here the Phone, Salary, and Start Date columns are using display formats so the data is easier to interpret:



ID	Name	Phone	Salary	Start Date
102	Fran	(617) 555-3985	\$45,700.00	Feb 26, 1995
105	Matthew	(617) 555-3840	\$62,000.00	Jul 2, 1992
129	Philip	(404) 555-2341	\$98,500.00	Aug 4, 1987
148	Julie	(617) 555-7835	\$51,432.00	Oct 4, 1991
160	Robert	(617) 555-3099	\$57,490.00	Dec 16, 1993
184	Melissa	(508) 555-2319	\$36,490.00	Apr 18, 1997

---

### Display formats not used for data entry

When you tab to a column containing a display format, InfoMaker removes the display format and displays the raw value for you to edit.

**FOR INFO** If you want to provide formatting used for data entry, you need to specify edit masks, as described in "Working with edit styles" on page 276.

---

## Using display formats

You work with display formats in the Database painter, the Table painter, the Form painter, and the Report painter.

What you do in the Database painter

In the Database painter, you can:

- ◆ Create, modify, and delete named display formats

The named display formats are stored in the repository. Once you define a display format, it can be used by any column of the appropriate data type in the database.

- ◆ Assign display formats to columns

These formats are used by default when you place the column in a report in the Report painter or a form in the Form painter.

What you do in the Table painter

In the Table painter, you can:

- ◆ Assign display formats to columns
- ◆ Remove display formats from columns

What you do in the Form and Report painters

In the Form and Report painters, you can:

- ◆ Accept the default display format assigned to a column in the Database or Table painter
- ◆ Override the default display format with another named format stored in the repository
- ◆ Create an ad hoc, unnamed format to use with one specific column

Display formats and the repository

Once you have placed a column in a report or form and have given it a display format (either the default format from the assignment made in the Database or Table painter for the column or a format assigned in the Form or Report painter), there is no longer any link to the named format in the repository.

If the definition of the display format later changes in the repository, the format for the column in a report or form will not change. If you want to use the modified format, you can reapply it to the column in the Form or Report painter.

## Working with display formats in the Database painter

Typically, you define display formats and associate them with columns in the Database painter, because display formats are properties of the data itself. Once you have associated a display format with a column in the Database painter, it is used by default each time the column is placed in a report or form.

---

### **Edit style takes precedence**

If a column has an associated edit style, the edit style takes precedence over a display format.

**FOR INFO** For more information, see "Working with edit styles" on page 276.

---

### **❖ To create or modify a display format:**

- 1 In the Database painter, open the table containing the column you want to apply a display format to.
- 2 Position the pointer on the column, select Properties from the column's popup menu, and select the Display tab.

All defined display formats for the corresponding data type are listed in the Display Format box. If the column has a display format defined, it is selected.

- 3 Click New to create a new format for the column.  
*or*  
Click Edit to modify the column's existing format.

- 4 If creating a new format, name it.
- 5 Define the display format using masks.

**FOR INFO** For information, see "Defining display formats" on page 269.

- 6 (Optional but recommended) Test the format by entering a test value and clicking the Test button.
- 7 Click OK to return to the Properties property sheet.
- 8 Click OK again to update the repository with your changes and to apply the format to the column.

You can use this display format with any column of the appropriate data type in the database.

---

**Another way to create and modify a display format**

Select Design>Display Format Maintenance from the menu bar to create and modify display formats.

When you create a display format this way, you are not creating the display format for a specific column and so you must tell InfoMaker what type of column will use the display format.

---

❖ **To apply a display format to a column:**

- 1 In the Database painter, position the pointer on the column, select Properties from the popup menu, and select the Display tab.  
All defined display formats for the corresponding data type display.
- 2 Select a format from the list in the Display Format box and click OK.  
The column now has the selected format associated with it in the repository.

---

**Another way to apply display formats**

In the Table painter, select a column and pick a display format from the Format dropdown listbox at the bottom of the workspace.

---

❖ **To remove a display format from a column:**

- 1 In the Database painter, position the pointer on the column, select Properties from the popup menu, and select the Display tab.
- 2 Click the highlighted format and click OK.  
The display format is no longer associated with the column.

---

**Another way to remove display formats**

In the Table painter, select a column and specify "(None)" in the Format dropdown listbox at the bottom of the workspace.

---

## **Working with display formats in the Report painter and Form painter**

Display formats you assign to a column in the Database painter are used by default when you place the column in a report or form. You can override the default format in the Report painter or Form painter by choosing another format from the repository or defining an ad hoc format for one specific column.

---

### About computed fields

You can assign display formats to computed fields using the same techniques as for columns in a table.

---

❖ **To specify a display format for a column in the Report painter or Form painter:**

- 1 In the Report painter or Form painter, move the pointer to the column, select Properties from the column's popup menu, and then select the Format tab.

Information appropriate to the data type of the selected column displays. The currently used format displays in the Format box. All formats for the data type defined in the repository are listed at the bottom.

- 2 Do one of the following:
  - ◆ Delete the display format.
  - ◆ Select a format in the repository from the list at the bottom of the dialog box.
  - ◆ Create a format for the column by typing it in the Format box. For more information, see "Defining display formats" next.

---

### Format not saved in the repository

If you create a format here, it is used only for the current column and is not saved in the repository.

---

- 3 Test the format by typing a value in the Test Value box and clicking Test.
- 4 Click OK to save your work.

---

### Shortcuts in the Report painter

To assign the Currency or Percent display format to a numeric column in a report, select the column, then click the Currency or Percent button in the PainterBar or select Edit>Format from the menu bar and choose the format from the cascading menu.

---

Customizing the toolbar

You can add buttons to the PainterBar that assign a specified display format to selected columns in reports and forms.

FOR INFO For more information, see Chapter 1, "Working with InfoMaker".



## Defining display formats

Display formats are represented through **masks**, where certain characters have special significance. InfoMaker supports four kinds of display formats, each using different mask characters:

- ◆ Numbers
- ◆ Strings
- ◆ Dates
- ◆ Times

For example, in a string format mask, each @ represents a character in the string and all other characters represent themselves. So you can use the following mask to display phone numbers:

`(@@@) @@@-@@@@`

### Combining formats

You can include different types of display format masks in a single format; use a space to separate the masks. For example, the following format section includes a date and time format:

`mmmm/dd/yyyy h:mm`

### Using sections

Each type of display format can have multiple sections, with each section corresponding to a form of the number, string, date, or time. Only one section is required; additional sections are optional and should be separated with semicolons (;).

The following format specifies different displays for positive and negative numbers (negative numbers will be displayed in parentheses):

`$$,##0;($$,##0)`

### Using keywords

Enclose display format keywords in square brackets. For example, you can use the keyword [General] when you want InfoMaker to determine the appropriate format for a number.

### Using colors

You can define a color for each display format section by specifying a color keyword before the format. The color keyword is the name of the color, or a number that represents the color, enclosed in square brackets: [RED] or [255]. The number is usually used only when a color is required that is not provided by name.

The named color keywords are:

[BLACK]            [MAGENTA]  
 [BLUE]            [RED]  
 [CYAN]            [WHITE]  
 [GREEN]           [YELLOW]

The formula for combining primary color values into a number is:

$$256*256*blue + 256*green + red=number$$

where the amount of each primary color is specified as a value from 0 to 255. For example, to specify cyan, substitute 255 for blue, 255 for green, and 0 for red. The result is 16776960.

The table below lists the blue, green, and red values you can use in the formula to create other colors:

Blue	Green	Red	Number	Color
0	0	255	255	Red
0	255	0	65280	Green
0	128	0	32768	Dark green
255	0	0	16711680	Blue
0	255	255	65535	Yellow
0	128	128	32896	Brown
255	255	0	16776960	Cyan
192	192	192	12632256	Light gray

Using special characters

To include a character in a mask that has special meaning in a display format, such as [, precede the character with a backslash (\). For example, to display a single quotation mark, enter \'.

## Number display formats

A number display format can have up to four sections, with only the first being required:

Positive-format;negative-format;zero-format>null-format

Special characters

Characters that have special meaning in number display formats are:

Character	Meaning
#	A number

Character	Meaning
0	A required number; a number will display for every 0 in the mask

Dollar signs, percent signs, decimal points, parentheses, and spaces display as entered in the mask.

#### Number keyword

You can use the following keywords as number display formats when you want InfoMaker to determine an appropriate format to use:

- ◆ [General]
- ◆ [Currency]

#### Examples

The following table shows how the values 5, -5, and .5 display when different format masks are applied:

Sample format	5	-5	.5
[General]	5	-5	0.5
0	5	-5	1
0.00	5.00	-5.00	0.50
#,##0	5	-5	1
#,##0.00	5.00	-5.00	0.50
\$\$,##0;(\$\$,##0)	\$5	(\$5)	\$1
\$\$,##0;-\$\$,##0	\$5	-\$5	\$1
\$\$,##0;[RED](\$\$,##0)	\$5	(\$5)	\$1
\$\$,##0.00;(\$\$,##0.00)	\$5.00	(\$5.00)	\$0.50
\$\$,##0.00;[RED](\$\$,##0.00)	\$5.00	(\$5.00)	\$0.50
0%	500%	-500%	50%
0.00%	500.00%	-500.00%	50.00%
0.00E+00	5.00E+00	-5.00E+00	5.00E-01

### String display formats

String display formats can have two sections. The first is required and contains the format for strings; the second is optional and specifies how to represent NULLs:

string-format;null-format

In a string format mask, each at-sign (@) represents a character in the string and all other characters represent themselves.

Example

This format mask:

**[red](@@@) @@@-@@@@**

displays the string 800YESCELT in red as:

(800) YES-CELT

## Date display formats

Date display formats can have two sections. The first is required and contains the format for dates; the second is optional and specifies how to represent NULLs:

date-format;null-format

Special characters

Characters that have special meaning in date display formats are:

Character	Meaning	Example
d	Day number with no leading zero	9
dd	Day number with leading zero if appropriate	09
ddd	Day name abbreviation	Mon
dddd	Day name	Monday
m	Month number with no leading zero	6
mm	Month number with leading zero if appropriate	06
mmm	Month name abbreviation	Jun
mmmm	Month name	June
yy	Two-digit year	97
yyyy	Four-digit year	1997

Colons, slashes, and spaces display as entered in the mask.

**About 2-digit years**

If you specify a 2-digit year in a report or form, InfoMaker assumes the date is the 20th century if the year is greater than or equal to 50. If the year is less than 50, InfoMaker assumes the 21st century. For example:

1/1/85 is interpreted as January 1, 1985.

1/1/40 is interpreted as January 1, 2040.

**Examples**

The following table shows how the date Friday, January 30, 1998, displays when different format masks are applied:

<b>Format</b>	<b>Displays</b>
[red]m/d/yy	1/30/98 in red
d-mmm-yy	30-Jan-98
dd-mmmm	30-January
mmm-yy	Jan-98
dddd, mmm d, yyyy	Friday, Jan 30, 1998

**Time display formats**

Time display formats can have two sections. The first is required and contains the format for times; the second is optional and specifies how to represent NULLs:

time-format;null-format

**Special characters**

Characters that have special meaning in time display formats are:

<b>Character</b>	<b>Meaning</b>
h	Hour with no leading zero (for example, 1)
hh	Hour with leading zero if appropriate (for example, 01)
m	Minute with no leading zero (must follow h or hh)
mm	Minute with leading zero if appropriate (must follow h or hh)
s	Second with no leading zero (must follow m or mm)
ss	Second with leading zero (must follow m or mm)

Character	Meaning
ffffff	Microseconds with no leading zeros. You can enter one to six f's; each f represents a fraction of a second (must follow s or ss)
AM/PM	Two-character, uppercase abbreviation (AM or PM as appropriate)
am/pm	Two-character, lowercase abbreviation (am or pm as appropriate)
A/P	One-character, uppercase abbreviation (A or P as appropriate)
a/p	One-character, lowercase abbreviation (a or p as appropriate)

Colons, slashes, and spaces display as entered in the mask.

---

**24-hour format is the default**

Times display in 24-hour format unless you specify AM/PM, am/pm, A/P, or a/p.

---

Time keyword

On Windows, you can use the following keyword as a time display format:

Keyword	Meaning
[Time]	The time format specified in the Microsoft Windows Control Panel
	<p><b>On Windows 3.1</b></p> <p>In addition to selecting the 12-hour radio button in the International Settings dialog box, you must also enter AM in the box on the 12-hour line and PM in the box on the 24-hour line</p>

Examples

The following table shows how the time 9:45:33.234567 PM displays when different format masks are applied:


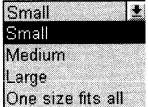
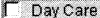



Format	Displays
h:mm AM/PM	9:45 PM
hh:mm A/P	09:45 P
h:mm:ss am/pm	9:45:33 pm
h:mm	21:45

<b>Format</b>	<b>Displays</b>
h:mm:ss	21:45:33
h:mm:ss:f	21:45:33:2
h:mm:ss:fff	21:45:33:234
h:mm:ss:ffffff	21:45:33:234567
m/d/yy h:mm	1/30/98 21:45

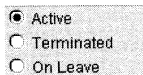
## Working with edit styles

You can define edit styles for columns. Edit styles specify how column data is presented in reports and forms. Unlike display formats, edit styles don't only affect the display of data; they also affect how you interact with the data in the Data Manipulation painter and in forms. Once you define an edit style, it can be used by any column of the appropriate data type in the database.

You can choose from the following edit styles:

Edit style	What the edit style does	Example
Edit box (default)	Displays a value in the box For data entry, type a value	
DropDownListBox	Displays a value from the dropdown listbox For data entry, select or enter a value	
CheckBox	Displays a checkbox selected or cleared For data entry, select or clear the checkbox	
RadioButtons	Displays radio buttons, one of which is selected For data entry, select one of the radio buttons	
Edit Mask	Displays formatted data For data entry, type a value	
DropDownDataWindow	Displays a value from a dropdown DataWindow For data entry, select a value	

For example, if you have a column Status that takes one of three values (A, T, or L), you might assign it the RadioButton edit style:





The Status data will be presented as radio buttons. You can simply click a button instead of having to type A, T, or L (and you don't have to create a validation rule to validate the typed input).

Here's a form that uses several edit styles: Edit Mask, DropDownListBox, CheckBox, and RadioButtons:

The screenshot shows a window titled "Employee Data" with the following fields and values:

Employee ID:	102	Birth Date:	06/05/1958
Manager ID:	501	Soc. Sec. No.:	017-34-9033
Emp. First Name:	Fran	Salary:	\$45,700.00
Emp. Last Name:	Whitney	Start Date:	02/26/1986
Department ID:	100	Termination Date:	00/00/0000
Street:	49 East Washington Street	Status:	<input checked="" type="radio"/> Active <input type="radio"/> Terminated <input type="radio"/> On Leave
City:	Needham	Health Insurance:	<input checked="" type="checkbox"/>
State:	MA	Life Insurance:	<input checked="" type="checkbox"/>
Zip Code:	02192	Day Care:	<input type="checkbox"/>
Phone:	(617) 555-3985		
Sex:	<input type="radio"/> Male <input checked="" type="radio"/> Female		

## Using edit styles

You work with edit styles in the Database painter, Table painter, Form painter, and Report painter.

What you do in the Database painter

In the Database painter, you can:

- ◆ Create, modify, and delete named edit styles

The edit styles are stored in the repository. Once you define an edit style, it can be used by any column of the appropriate data type in the database.

- ◆ Assign edit styles to columns

These styles are used by default when you place the column in a report in the Report painter or in a form in the Form painter.

What you do in the Table painter

In the Table painter you can:

- ◆ Assign edit styles to columns and remove them from columns

What you do in the Form and Report painters

In the Form and Report painters, you can:

- ◆ Accept the default edit style assigned to a column in the Database or Table painter
- ◆ Override the default edit style with another named style stored in the repository
- ◆ Create an ad hoc, unnamed edit style to use with one specific column

Edit styles and the repository

Once you have placed a column in a form or report and have given it an edit style (either the default style from the assignment made in the Database painter for the column or a style assigned in the Report painter or Form painter), InfoMaker records the name and definition of the edit style in the form or report.

However, if the definition of the edit style later changes in the repository, the edit style for the column in a form or report will not change automatically. You can update the column by reassigning the edit style to it in the form or report.

## Working with edit styles in the Database Painter

Typically, you define edit styles in the Database painter, because edit styles are properties of the data itself. Once defined in the Database painter, the styles are used by default each time the column is placed in a report or form.

### ❖ To create or modify an edit style:

- 1 In the Database painter, position the pointer on the column, select Properties from the popup menu, then select the Edit Style tab.  
  
All defined edit styles are listed in the Style Names box. If the column has an edit style defined, it is selected.
- 2 Click New to create a new edit style.  
*or*  
Click Edit to modify an existing edit style.
- 3 If creating a new edit style, select the edit style type from the Style dropdown listbox.
- 4 Specify the properties of the edit style.  
  
FOR INFO For information, see "Defining edit styles" on page 280.
- 5 Click OK to return to the Edit Style property page.
- 6 Click OK again to apply the edit style to the column and to store it in the repository.

---

**Another way to create and modify an edit style**

Select Design>Edit Style Maintenance from the menu bar to create an edit style (independent of a column), or to modify or delete an existing edit style.

---

You can use the new or modified edit style with any column of the appropriate data type in the database.

❖ **To apply an existing edit style to a column:**

- 1 In the Database painter, position the pointer on the column, select Properties from the popup menu, then select the Edit Style tab.
- 2 Select a style for the appropriate data type and click OK.  
InfoMaker associates the selected edit style with the column in the repository.

---

**Another way to apply edit styles**

In the Database painter, select Definition from the column's popup menu to display the Table painter. Then select a column and pick an edit style from the Edit dropdown listbox at the bottom of the workspace.

---

❖ **To remove an edit style from a column:**

- 1 In the Database painter, position the pointer on the column, select Properties from the popup menu, then select the Edit Style tab.
- 2 Click the highlighted edit style and click OK.  
The edit style is no longer associated with the column.

---

**Another way to remove edit styles**

In the Table painter, select a column and specify "(None)" in the Edit dropdown listbox at the bottom of the workspace.

---

## **Working with an edit style for a column in the Form or Report painter**

Edit styles you assign to a column in the Database painter are used by default when you place the column in a form or report. You can override the edit style in the Form or Report painter by choosing another edit style from the repository or defining an ad hoc style for one specific column.

❖ **To specify an edit style for a column:**

- 1 In the Form or Report painter, move the pointer to the column, select Properties from the column's popup menu, and then select the Edit tab.
- 2 Select the type of edit style you want from the Style dropdown listbox.  
The information on the Edit property page changes to be appropriate to the type of edit style you selected.
- 3 Do one of the following:
  - ◆ Select an edit style from the Name box.
  - ◆ Create an ad hoc edit style for the column, as described in "Defining edit styles" next.
- 4 Click OK to override the associated edit style.

## Defining edit styles

This section describes how to specify each type of edit style.

### The Edit edit style

By default, columns use the Edit edit style, which displays data in an edit control. You can customize the appearance and behavior of the edit control by modifying a column's Edit edit style. To do so, select Edit in the Style dropdown listbox and specify the properties for that style:

- ◆ To restrict the number of characters you can enter, enter a value in the Limit box.
- ◆ To convert the case of characters upon display, enter an appropriate value in the Case box.
- ◆ To have entered values display as asterisks for sensitive data, check the Password box.
- ◆ To allow you to tab to the column but not change the value, check the Display Only box.
- ◆ To define a code table to determine which values are displayed and which values are stored in the database, check the Use Code Table box and enter display and data values for the code table.

FOR INFO See "Defining a code table" on page 288.

❖ **To use the Edit edit style:**

- 1 Select Edit from the Style box, if it is not already selected.
- 2 Select the properties you want.

**The DropDownListBox edit style**

You can use the DropDownListBox edit style to have columns display as dropdown listboxes when you run a form:



Typically, this edit style is used with code tables, where you can specify display values (which you see) and shorter data values (which are stored in the database).

In the DropDownListBox edit style, the display values of the code table display in the ListBox portion of the DropDownListBox. The data values are the values that are sent to the database when it is updated.

In the preceding example, when you see the value Business Services, the corresponding data value could be 200.

❖ **To use the DropDownListBox edit style:**

- 1 Select DropDownListBox from the Style box.
- 2 Select the appropriate properties from the Options area.
- 3 Enter the value you want to appear in the Display Value box and the corresponding data value in the Data Value box.
- 4 Click the Add, Delete, and Insert buttons to modify the values as appropriate, then click OK when you are done.

For more information

For more about code tables, see "Defining a code table" on page 288.

## The CheckBox edit style

If a column can take only one of two (or perhaps three) values, you might want to display the column as a checkbox; you can select or clear the checkbox to specify a value. In the following entry from a form, you can simply check or clear a box to indicate whether an employee has health insurance:

**Health Insurance:**

### ❖ To use the CheckBox edit style:

- 1 Enter in the Text box the text you want displayed next to the checkbox.

---

#### Using accelerator keys

If the CheckBox has an accelerator key, enter an ampersand (&) before the letter in the text that represents the accelerator key.

---

- 2 Enter in the Data Value For box the values you want to store in the database when the CheckBox is checked (on) or unchecked (off).  
If you selected the 3 States box, an optional third state box (other) appears, for the case when the condition is neither on nor off.
- 3 Click OK when done.

What happens

The value you enter in the Text box becomes the display value, and values entered for On, Off, and Other become the data values.

When you check or clear the checkbox to modify the data, InfoMaker enters the appropriate data value in its buffer. When you save your changes, InfoMaker sends the corresponding data values to the database.

## Centering checkboxes without text

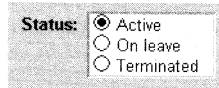
You may find it useful to center checkboxes used for columns of information. First make the text object used for the column header and the column object the same size and left aligned. Then you can center the checkboxes and the column header.

### ❖ To center checkboxes without text:

- 1 In the edit style (checkbox) property page for the column, make sure the Left Text checkbox is not selected and that the Text box where you specify associated text is empty.
- 2 Select the column object and specify centering using the Stylebar or by selecting Properties>General>Alignment>Center from the menu bar.

## The RadioButtons edit style

If a column can take one of a small number of values, you might want to display the column as radio buttons:



### ❖ To use the RadioButtons edit style:

- 1 Specify how many radio buttons will display in the Columns Across box.
- 2 Enter a set of display and data values for each button you want to display.

The display values you enter become the text of the buttons; the data values are stored in the database.

---

### Using accelerator keys

To use an accelerator key on a radio button, enter an ampersand (&) in the Display Value before the letter that will be the accelerator key.

---

- 3 Click OK when done.

What happens

You select values by clicking a radio button. When you save your changes, InfoMaker sends the corresponding data values to the database.

## The EditMask edit style

Sometimes you need to enter data that has a fixed format. For example, in North America phone numbers have a 3-digit area code, followed by three digits, followed by four digits. You can define an edit mask that specifies the format to make it easier for you to enter values:

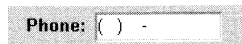


Edit masks consist of special characters that determine what can be entered in the column. They can also contain punctuation characters to aid you.

For example, to make it easier for you to enter phone numbers in the proper format, specify this mask:

**(###) ###-####**

When you insert a row in a form, the punctuation characters display in the box and the cursor jumps over them as you type:



#### Special characters

Special characters used in edit masks are the same ones used in display formats.

**FOR INFO** For information about special characters used in masks, see "Defining display formats" on page 269.

#### Keyboard behavior

Note the following about how certain keystrokes behave in edit masks:

- ◆ Both BACKSPACE and SHIFT-BACKSPACE delete the preceding character
- ◆ DELETE deletes everything that is selected
- ◆ Non-numeric edit masks treat any characters that don't match the mask pattern as delimiters

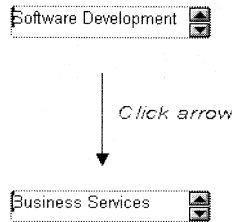
Also, note certain behavior in the Date, Datetime, and Time edit masks:

- ◆ A nonzero number in the first position of dd, mm, hh, ss, or mm (minutes) followed by the delimiter character, will cause the typed digit to be preceded by 0. For example, typing **1/1/97** results in **01/01/97**.
- ◆ A number in the first position that is greater than the maximum day, month, hour, and so on, causes the last-typed number to be entered as the second position and the first position to be set to 0. The cursor is positioned in front of the second position. For example, entering **959595** results in **05/05/95**.
- ◆ The strings 00/00/00 or 00/00/0000 are interpreted as the NULL value for the column. If you enter 00 for either the month or the day, the other fields must also be zero or the date will be rejected as invalid.



## Using spin controls

You can define an edit mask as a **spin control**, a box that contains up and down arrows that you can click to cycle through fixed values. For example, you can set up a code table that provides the valid entries in a column; you simply click an arrow to select an entry. Used this way, a spin control works like a dropdown listbox that displays one value at a time:



**FOR INFO** For more about code tables, see "Defining a code table" on page 288.

❖ **To use an EditMask edit style:**

- 1 Select EditMask in the Style box if it not already selected.
- 2 Define the mask in the Mask box. Click the special characters in the Masks box to use them.
- 3 Specify other properties for the edit mask.
- 4 (Optional) Test the mask by typing a value in the Test box.
- 5 Click OK when done.

When use your EditMask, check its appearance and behavior. If characters do not appear as you expect, you may want to change the font size or the size of the EditMask.

## The DropDownDataWindow edit style

Sometimes another data source determines which data is valid for a column.

Consider this situation: the Department table includes two columns, Dept\_id and Dept\_name, to record your company's departments. The Employee table records your employees. The Department column in the Employee table can have any of the values in the Dept\_id column in the Department table.

As new departments are added to your company, you want the form containing the Employee table to automatically provide the new departments as choices when you enter values in the Department column.

In situations such as these, you can specify the DropDownDataWindow edit style for a column: it is populated from another report. When you go to the column, the contents of the DropDownDataWindow display, showing the latest data:

<b>Department:</b>	Business Services	▼
	<b>Name</b>	<b>Code</b>
<b>Status:</b>	Software Development	100
	Business Services	200
	Corporate Marketing	300
<b>Start date:</b>	Marketing	400
<b>Expiration date:</b>		

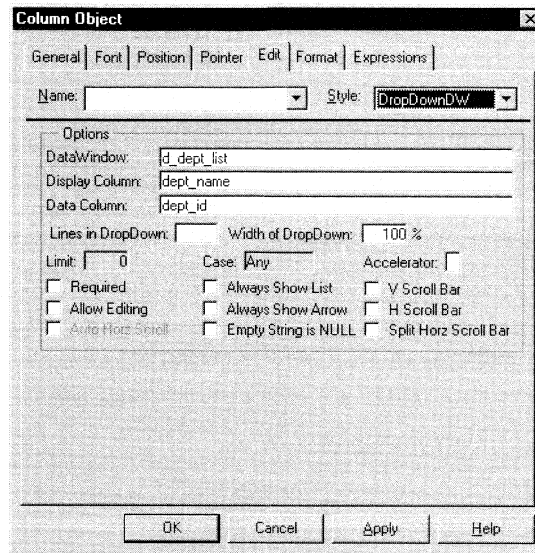
❖ **To use the DropDownDataWindow edit style:**

- 1 Create a report that contains the columns in the detail band whose values you want to use in the column.

You will often choose at least two columns: one column that contains values that the user sees and another column containing values to be stored in the database. In the example above, you would create a report containing the DeptID and DeptName columns in the Department table. Assume this report is named d\_dept\_list.

- 2 For the column getting its data from the report, select the DropDownDataWindow edit style.

In the example, you would specify the DropDownDataWindow edit style for the Department column in the Employee table:



- 3 In the DataWindow box, select the report that contains the data for the column from the dropdown list (in the example, d\_dept\_list). (When you click the edit box, you will see the list.)
- 4 In the Display Column box, select the column containing the values that will display in the report or form (in the example, DeptName).
- 5 In the Data Column box, select the column containing the values that will be stored in the database (in the example, DeptID).
- 6 Specify other properties for the edit style and click OK when done.

#### What happens

When you preview a table's data (or run a form) and data is retrieved, a column with the DropDownDataWindow edit style is populated, providing data to the DropDownDataWindow.

When you run a form, go to the column, and drop it down, the entire report displays. When you select a display value and update the database, the corresponding data value is stored in the database.

---

### **Limit on size of data value**

The data value for a column that uses the DropDownDataWindow edit style is limited to 511 characters.

---

## **Defining a code table**

To reduce storage needs, you frequently want to store short, encoded values in the database. But these encoded values may not be meaningful to users. To make reports and forms easy to use, you can define **code tables**.

Each row in a code table is a pair of corresponding values: a display value and a data value. The display values are those you see in a report and in a form. The data values are those that are saved in the database.

## **How code tables are implemented**

You can define a code table as a property of the following column edit styles:

- ◆ Edit
- ◆ DropDownListBox
- ◆ RadioButtons
- ◆ DropDownDataWindow
- ◆ EditMask, using spin control

The steps to specify the code table property for each edit style are similar: you begin by defining a new edit style in the Database painter. Once you select an edit style, use the following specific procedure to define the code table property.

**FOR INFO** For how to create an edit style, see "Working with edit styles" on page 276.

---

### **Allowing NULL values**

An internal InfoMaker code, NULL!, indicates NULL values are allowed. To use this code, specify NULL! as the data value, then specify a display format for NULLs for the column.

---

❖ **To define a code table as a property of the Edit edit style:**

- 1 Select the Use Code Table checkbox.
- 2 Enter the display and data values for the code table.
- 3 If you want to restrict input in the column to values in the code table, select the Validate Using Code Table checkbox.  
  
FOR INFO For more information, see "Validating user input" on page 290.
- 4 Click OK to accept the edit style definition.

❖ **To define a code table as a property of the DropDownListBox edit style:**

- 1 Enter the display and data values for the code table.
- 2 If you want to restrict input in the column to values in the code table, clear the Allow Editing checkbox.  
  
FOR INFO For more information, see "Validating user input" on page 290.
- 3 Click OK to accept the edit style definition.

❖ **To define a code table as a property of the RadioButtons edit style:**

- 1 Enter the display and data values for the code table.
- 2 Click OK to accept the edit style definition.

❖ **To define a code table as a property of the DropDownDataWindow edit style:**

- 1 Specify the column that provides the display values in the Display Column box.
- 2 Specify the column that provides the data values in the Data Column box.
- 3 If you want to restrict input to values in the code table, clear the Allow Editing checkbox.
- 4 Click OK to accept the edit style definition.

❖ **To define a code table as a property of the EditMask edit style:**

- 1 Select the Spin Control checkbox.
- 2 Select the Code Table checkbox.
- 3 Enter the display and data values for the code table.
- 4 Click OK to accept the edit style definition.

## How code tables are processed

When data is retrieved into a report or form column with a code table, processing begins at the top of the data value column. If the data matches a data value, the corresponding display value displays. If there is no match, the actual value displays.

Consider this example:

Display values	Data values
Massachusetts	MA
Massachusetts	ma
ma	MA
Mass	MA
Rhode Island	RI
RI	RI

If the data is MA or ma, the corresponding display value (Massachusetts) displays. If the data is Ma, there is no match, so Ma displays.

---

### Case sensitivity

Code table processing is case sensitive.

---

If the code table is in a DropDownList edit style, and if the column has a code table that contains duplicate display values, then each value displays only once. So if this code table is defined for a column in a form that has a DropDownList edit style, Massachusetts and Rhode Island display in the ListBox portion of the DropDownList.

## Validating user input

When you enter data into a column in a form, processing begins at the top of the display value column of the associated code table.

If the data matches a display value, the corresponding data value is put in the internal buffer. For each display value, the first data value is used. Using the sample code table, if you enter Massachusetts, ma, or Mass, MA is the data value.

You can specify that *only* the values in the code table are acceptable:

- ◆ For a column using the Edit edit style, select the Validate Using Code Table checkbox.
- ◆ For the DropDownListBox and DropDownDataWindow edit styles, clear the Allow Editing checkbox: you cannot type a value.

---

**Code table data**

The data values in the code table must pass validation for the column and must have the same data type as the column.

---

## Working with validation rules

When you enter data in a form, you want to be sure the data is valid before using it to update the database. One way to do this is through validation rules.

You usually define validation rules in the Database painter. To use a validation rule, you associate it with a column in the Database painter, Table painter, or form painter.

InfoMaker uses validation rules when you enter data:

- ◆ Directly in the database using the Data manipulation painter
- ◆ In a form

---

### Another technique

You can also perform data validation through code tables, which are implemented through a column's edit style.

**FOR INFO** For more information, see "Working with edit styles" on page 276.

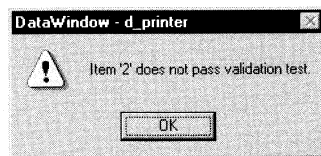
---

## Understanding validation rules

Validation rules are criteria that a form uses to validate data entered into a column. They are Powersoft-specific and therefore not enforced by the DBMS. Validation rules apply to forms (which support updating) but not to reports.

Validation rules assigned in the Database painter are used by default when you place columns in a form. You can override the default rules in the Form painter.

A validation rule is an expression that evaluates to either TRUE or FALSE. If the expression evaluates to TRUE for an entry into a column, InfoMaker accepts the entry. If the expression evaluates to FALSE, the entry is not accepted and an error message is displayed:



You can customize the message displayed when a value is rejected.



## Using validation rules

You work with validation rules in the Database painter, Table painter, and Form painter.

What you do in the Database painter

In the Database painter, you can:

- ◆ Create, modify, and delete named validation rules

The validation rules are stored in the repository. Once you define a validation rule, it can be used by any column of the appropriate data type in the database.

- ◆ Assign validation rules to columns

These rules are used by default when you place the column in a form in the Form painter.

What you do in the Table painter

In the Table painter, you can:

- ◆ Assign validation rules to columns and remove them from columns

What you do in the Form painter

In the Form painter, you can:

- ◆ Accept the validation rule a column was given in the Database painter or Table painter
- ◆ Create an ad hoc, unnamed rule to use with one specific column

Validation rules and the repository

Once you have placed a column that has a validation rule from the repository in a form, there is no longer any link to the named rule in the repository.

If the definition of the validation rule later changes in the repository, the rule for the column in a form will not change.

## Defining validation rule properties in the Database painter

Typically, you will define validation rules in the Database painter, because validation rules are properties of the data itself. Once defined in the Database painter, the rules are used by default each time the column is placed in a form.

This section describes the ways you can manipulate validation rules in the Database painter.

❖ **To create or modify a validation rule:**

- 1 In the Database painter, position the pointer on the column, display the column's popup menu, select Properties from the popup menu, then select the Validation tab.

All defined validation rules for the corresponding data type display. If the column has a validation rule associated with it, it is selected.

- 2 Click New to create a new rule.  
*or*  
Click Edit to modify an existing rule.
- 3 If creating a new rule, name it.
- 4 Define the expression for the validation rule.  
FOR INFO For information, see "Defining the expression" on page 295.
- 5 (Optional) Customize the error message.  
FOR INFO For information, see "Customizing the error message" on page 296.
- 6 Click OK to return to the Column property sheet.
- 7 Click OK again to apply the validation rule to the column and to store it in the repository.

You can use this rule with any column of the appropriate data type in the database.

---

**Another way to create and modify a validation rule**

Select Design>Validation Maintenance from the menu bar to create a new validation rule (independent of a column), or modify or delete an existing rule.

FOR INFO For information, see "Maintaining the entities" on page 300.

---

❖ **To apply an existing validation rule to a column:**

- 1 In the Database painter, position the pointer on the column, display the column's popup menu, select Properties, and select the Validation tab.

All defined validation rules for the corresponding data type display. If the column has a validation rule associated with it, it is selected.

- 2 Select a rule and click OK.

The column now has the selected rule associated with it in the repository. Whenever you use this column in a form, it will use this validation rule unless you override it in the Form painter.

**Another way to apply a validation rule to a column**

Select Definition from a table's popup menu, then select a column, and pick a validation rule from the Validation dropdown listbox in the Extended Attributes groupbox.

❖ **To remove a validation rule from a column:**

- 1 In the Database painter, position the pointer on the column, display the column's popup menu, select Properties, and select the Validation tab.

All defined validation rules for the corresponding data type display with the validation rule associated with the column highlighted.

- 2 Click the highlighted rule and click OK.

**Defining the expression**

A validation rule is a boolean expression. InfoMaker applies the boolean expression to an entered value. If the expression returns TRUE, the value is accepted. Otherwise, the value is not accepted and an error message displays.

What expressions  
can contain

You can use any valid InfoMaker expression in validation rules.

InfoMaker functions are displayed in the Functions listbox and can be pasted into the definition.

**FOR INFO** For information about these functions, see Chapter 23, "DataWindow Painter and InfoMaker Functions".

Use the notation *@placeholder* (where *placeholder* is any group of characters) to indicate the current column in the rule. When you define a validation rule in the Database painter, InfoMaker stores it in the repository with the placeholder name. During execution, InfoMaker substitutes the value of the column for *placeholder*.

Pasting the  
placeholder

The name of the column for which you initially create the rule is frequently used as the placeholder. A button in the Paste area is labeled with the column name preceded by an @. You can click the button to paste the placeholder into the validation rule.

An example

For example, to make sure that both Age and Salary are greater than zero using a single validation rule, define the validation rule for one of the columns. If defining it for Salary, the expression would be:

```
@salary > 0
```

Then use the same validation rule for Age. At execution time, InfoMaker substitutes the appropriate values for the column data when the rule is applied.

## Using match values for character columns

If the column you are defining the validation rule for is a character column, the Match button is active, allowing you to match the contents of the column to a specified text pattern (for example, `^[0-9]+$` for all numbers and `^[A-Za-z]+$` for all letters).

### ❖ To specify a match pattern for character columns:

- 1 Click the Match button.  
The Match Pattern dialog box displays.
- 2 Enter the text pattern you want to match the column to.  
*or*  
Select a displayed pattern.
- 3 (Optional) Enter a test value and click the Test button to test the pattern.
- 4 Click OK when you are satisfied that the pattern is correct.

FOR INFO For more on the Match function and text patterns, see Chapter 23, "DataWindow Painter and InfoMaker Functions".

## Customizing the error message

When you define a validation rule, InfoMaker automatically creates the error message that displays by default when you enter an invalid value:

```
'Item ~' + @ ColumnName + '~' does not pass validation test.'
```

You can edit the string expression in the Validation Error Message textbox to create a custom error message.

---

### Different syntax in the form painter

If you're working in the form painter, you can enter a string expression in the Error Message Expression textbox, but you don't use the @ sign for placeholders. For example, this is the default message:

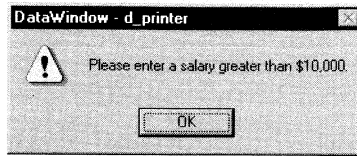
```
'Item ~' + ColumnName + '~' does not pass validation test.'
```

---

A validation rule for the Salary column in the Employee table might have the following custom error message associated with it:

'Please enter a salary greater than \$10,000.'

If you enter a salary less than or equal to \$10,000, the custom error message displays:



## Specifying initial values

As part of defining a validation rule, you can supply an initial value for a column.

### ❖ To specify an initial value for a column:

- 1 Select Properties from the column's popup menu and select the Validation tab.
- 2 Specify a value in the Initial Value box.

## Defining a validation rule in the Form painter

Validation rules you assign to a column in the Database or Table painter are used by default when you place the column in a form. You can override the validation rule in the Form painter by defining an ad hoc rule for one specific column.

### ❖ To specify a validation rule for a column:

- 1 In the Form painter, select Properties from the column's popup menu, then select the Validation tab.

The Column Object property sheet displays.

- 2 Create or modify the validation expression, as described next.
- 3 (Optional) Enter a string or string expression to customize the validation error message.

**FOR INFO** For more information, see "Customizing the error message" on page 296.

- 4 Click Verify to verify the syntax.
- 5 Click OK.

---

**Used for current column only**

If you create a validation rule here, it is used only for the current column and is not saved in the repository.

---

## Specifying the expression

Since you might just have entered a value in the column, validation rules refer to the current data value, which you can obtain through the `GetText` function.

Using `GetText` ensures that the most recent data entered in the current column is evaluated.

---

**InfoMaker does the conversion for you**

If you have associated a validation rule for a column in the Database painter, InfoMaker automatically converts the syntax to use `GetText` when you place the column in a form.

---

`GetText` returns a string. Be sure to use a data conversion function (such as `Integer` or `Real`) if you want to compare the entered value with a data type other than string.

**FOR INFO** For more on the `GetText` function and text patterns, see Chapter 23, "DataWindow Painter and InfoMaker Functions".

Referring to other columns

You can refer to the values in other columns by specifying their names in the validation rule. You can paste the column names in the rule using the Columns box.

## Examples

Here are some examples of validation rules.

**Example 1** To check that the data entered in the current column is a positive integer, use this validation rule:

```
Integer(GetText( )) > 0
```

**Example 2** If the current column contains the discounted price and the column named `Full_Price` contains the full price, you could use the following validation rule to evaluate the contents of the column using the `Full_Price` column:

```
Match(GetText( ), "^[0-9]+$") AND
```

```
Real(GetText( )) < Full_Price
```

To pass the validation rule, the data must be all digits (must match the text pattern `^[0-9]+$`) and must be less than the amount in the `Full_Price` column.

Notice that to compare the numeric value in the column with the numeric value in the `Full_Price` column, the `Real` function was used to convert the text to a number.

**Example 3** In your company, a product price and a sales commission are related in the following way:

- ◆ If the price is greater than or equal to \$1000, the commission is between 10 percent and 20 percent
- ◆ If the price is less than \$1000, the commission is between 4 percent and 9 percent

The `Sales` table has two columns, `Price` and `Commission`. The validation rule for the `Commission` column is:

```
(Number(GetText( )) >= If(price >= 1000, .10, .04))
AND
(Number(GetText( )) <= If(price >= 1000, .20, .09))
```

A customized error message for the `Commission` column is:

```
"Price is " + if(price >= 1000,
"greater than or equal to","less than") +
" 1000. Commission must be between " +
If(price >= 1000, ".10", ".04") + " and " +
If(price >= 1000, ".20.", ".09.")
```

## Maintaining the entities

InfoMaker provides facilities you can use to create, modify, and delete display formats, edit styles, and validation rules independently of their association with columns.

❖ **To maintain display formats, edit styles, and validation rules:**

- 1 Open the Database painter.
- 2 Select the appropriate entry from the Design menu:
  - ◆ Edit Style Maintenance
  - ◆ Display Format Maintenance
  - ◆ Validation Maintenance

A dialog box displays listing all the corresponding entities that are in the repository.

- 3 Do one of the following:
  - ◆ To create a new entity, click New.
  - ◆ To modify an existing entity, select it, then click Edit.
  - ◆ To delete an existing entity, select it, then click Delete.

---

**Caution**

If you delete a display format, edit style, or validation rule, it is removed from the repository. Columns in the database are no longer associated with the entity.

---



# Filtering, Sorting, and Grouping Rows

## About this chapter

This chapter describes how you can customize your report by doing the following in the Report painter:

- ◆ Defining filters to limit which of the retrieved rows are displayed in the report
- ◆ Sorting rows after they have been retrieved from the database
- ◆ Displaying the rows in groups and calculating statistics on each group

## Contents

<b>Topic</b>	<b>Page</b>
Filtering rows	302
Sorting rows	305
Grouping rows	308

## Filtering rows

You can use WHERE and HAVING clauses and retrieval arguments in the SQL SELECT statement for the report to limit the data that is retrieved from the database. This reduces retrieval time and space requirements for the report data.

However, you may want to further limit the data that displays in the report. For example, you might want to:

- ◆ Retrieve many rows and initially display only a subset
- ◆ Limit the data that is displayed using InfoMaker functions (such as If) that are not valid in the SELECT statement

### Using filters

In the Report painter, you can define filters, which will limit the rows that display when you run the report. Filters can use most InfoMaker functions.

---

#### Filters don't affect which rows are retrieved

A filter operates against the retrieved data. It does not re-execute the SELECT statement.

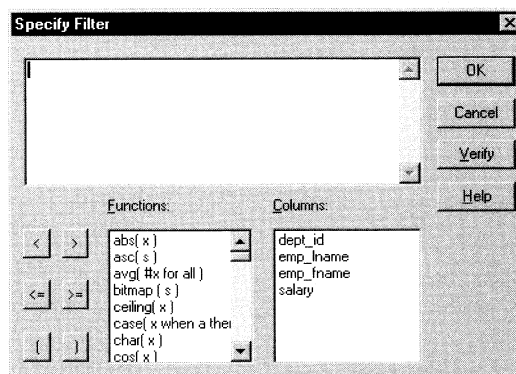
---

### Defining a filter

#### ❖ To define a filter:

- 1 In the Report painter, select Rows>Filter from the menu bar.

The Specify Filter dialog box displays:



- In the Specify Filter dialog box, enter a boolean expression that InfoMaker will test against each retrieved row.

If the expression evaluates to TRUE, the row will be displayed. You can specify any valid expression in a filter. You can paste commonly used functions, names of columns, computed fields, retrieval arguments, and operators into the filter.

---

#### International considerations

So that anything you create with InfoMaker will run the same no matter in which country it is deployed, filter expressions require U.S. notation for numbers. That is, comma always represents the thousands delimiter and period always represents the decimal place when you specify expressions in the InfoMaker environment.

---

- (Optional) Click Verify to make sure the expression is valid.
- Click OK to return to the workspace.
- (Optional) Test the filter by clicking Preview.  
Only rows meeting the filter criteria will be displayed.

---

#### Filtered rows and updates

Filtered rows are updated when you update the database.

---

#### Removing a filter

##### ❖ To remove a filter:

- Select Rows>Filter from the menu bar.
- Delete the filter expression from the Specify Filter dialog box, then click OK.

#### Examples of filters

Assume that a report retrieves employee rows. Three of the columns are Salary, Status, and Emp\_Lname:

To display these rows	Use this filter
Employees with salaries over \$50,000	Salary > 50000
Active employees	Status = 'A'
Active employees with salaries over \$50,000	Salary 50000 AND Status = 'A'

**To display these rows**

Employees whose last names  
begin with H

**Use this filter**

```
left(Emp_Lname, 1) = 'H'
```

## Sorting rows

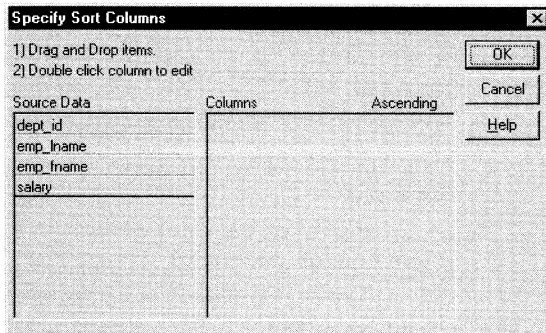
You can use an `ORDER BY` clause in the SQL `SELECT` statement for the report to sort the data that is retrieved from the database. If you do this, the DBMS itself does the sorting and the rows are brought into InfoMaker already sorted.

However, you might want to sort the rows after they are retrieved. For example you might want to:

- ◆ Offload the processing from the DBMS
- ◆ Sort on an expression, which is not allowed in the `SELECT` statement but is allowed in InfoMaker

### ❖ To sort the rows:

- 1 Select Rows>Sort from the menu bar:

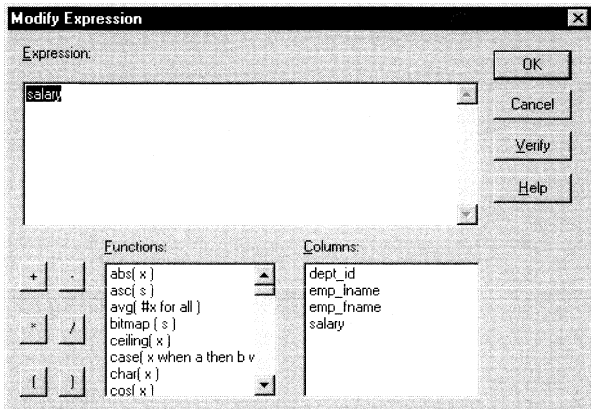


- 2 Drag the columns that you want to sort the rows on to the Columns box, and specify whether you want to sort in ascending or descending order.

The order of the columns determines the precedence of the sort. To reorder the columns, drag them up or down in the list. To delete a column from the sort columns list, drag the column outside the dialog box.

- 3 You can also specify expressions to sort on: for example, if you have two columns, Revenues and Expenses, you can sort on the expression *Revenues – Expenses*.

To specify an expression to sort on, double-click a column name in the Columns box, modify the expression in the Modify Expression dialog box, and click OK:



You return to the Specify Sort Columns dialog box with the expression displayed.

---

#### **If you change your mind**

You can remove a column or expression from the sorting specification by simply dragging it and releasing it outside the Columns box.

---

- 4 Click OK when you have specified all the sort columns and expressions.

## **Suppressing repeating values**

When you sort on a column, you may have several rows with the same value in one column. In this case, you may want to suppress the repeating values in that column.

When you suppress a repeating value, the value displays at the start of each new page and, if you are using groups, each time a value changes in a higher group.

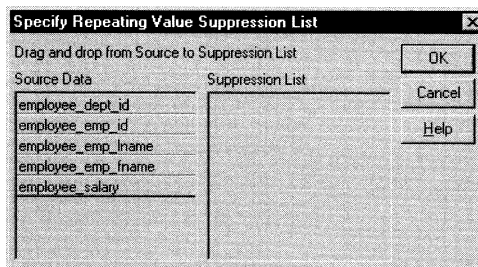
For example, if you have sorted employees by department ID, you might want to suppress all but the first occurrence of each department ID in the report:

Dept	ID	Name	Salary
300	390	Davidson , Jo Ann	\$57,090
	586	Coleman , James	\$42,300
	757	Higgins , Denis	\$43,700
	879	Coe , Kristen	\$36,500
	1293	Shea , Mary Anne	\$138,948
	1336	Bigelow , Janet	\$31,200
	1390	Lilton , Jennifer	\$58,930
	1483	Letiecq , John	\$75,400
400	184	Espinoza , Melissa	\$36,490
	207	Francis , Jane	\$53,870
	318	Crow , John	\$41,701
	409	Weaver , Bruce	\$46,550
	591	Barletta , Irene	\$45,450
	888	Charlton , Doug	\$28,300
	992	Butterfield , Joyce	\$34,011

❖ **To suppress repeating values:**

- 1 Select Rows>Suppress Repeating Values from the menu bar.

The Specify Repeating Value Suppression List dialog box displays:



- 2 Drag the columns whose repeated values you want to suppress from the Source Data box to the Suppression List box and click OK.

**If you change your mind**

You can remove a column from the suppression list by simply dragging it and releasing it outside the Suppression List box.

## Grouping rows

You can group related rows together and, optionally, calculate statistics for each group separately. For example, you might want to group employee information by department and get total salaries for each department.

How groups are defined

Each group is defined by one or more report columns. Each time the value in a grouping column changes, a **break** occurs and a new section begins.

For each group you can:

- ◆ Display the rows in each section
- ◆ Specify the information you want displayed at the beginning and end of each section
- ◆ Specify page breaks after each break in the data
- ◆ Reset the page number after each break



Grouping example

The following report retrieves employee information. It has one group defined, Dept\_ID. So it groups rows into sections according to the value in the Dept\_ID column. In addition, it displays:

- ◆ Department ID before the first row for that department
- ◆ Totals and averages for salary and salary plus benefits (a computed column) for each department
- ◆ Grand totals for the company at the end

Total Compensation Report								Page 3 of 3	
Salary Plus Benefits								01/12/96	
<i>Value of health ins. - \$4,000</i> <i>Value of life insurance - \$25,000 salary \$1,000</i> <i>Value of day care - \$5,200</i>									
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits	
400	1507	Ruth	Wetherby	\$35,745	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$40,739	
	1576	Scott	Evanr	\$68,940	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$74,114	
	1607	Mark	Marris	\$61,300	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$66,433	
	1643	Elizabeth	Lambert	\$29,384	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$34,344	
	1684	Janet	Hildebrand	\$45,829	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,878	
	1740	Robert	Nielson	\$34,889	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$39,878	
	1751	Alex	Ahmed	\$34,992	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$39,982	
	<b>Total:</b>				<b>\$698,251</b>				<b>Total:</b>
<b>Average:</b>				<b>\$43,641</b>				<b>Average:</b>	<b>\$49,028</b>
500	191	Jeannette	Bertrand	\$29,800	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$39,962	
	703	Jane	Martinez	\$55,501	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,602	
	750	Jane	Braun	\$34,300	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$39,286	
	868	Felicia	Kua	\$28,200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$33,153	
	921	Charles	Crawley	\$41,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$46,726	
	1013	Jaroph	Barker	\$27,290	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$27,438	
	1570	Anthony	Robeira	\$34,576	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$39,564	
	1615	Sheila	Ramora	\$27,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$37,649	
1658	Michael	Lynch	\$24,903	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$29,838		
<b>Total:</b>				<b>\$303,770</b>				<b>Total:</b>	<b>\$354,219</b>
<b>Average:</b>				<b>\$33,752</b>				<b>Average:</b>	<b>\$39,358</b>
<b>Grand total:</b>				<b>\$3,749,147</b>				<b>Grand total:</b>	<b>\$4,170,705</b>
<b>Overall average:</b>				<b>\$49,989</b>				<b>Overall average:</b>	<b>\$55,609</b>

How to do it

You can create a grouped report two ways:

- ◆ Use the Group presentation style to create a grouped report from scratch.
- ◆ Take an existing tabular report and define grouping in the Report painter workspace ("Defining groups in an existing report" on page 314).

## Using the Group presentation style

One of the report presentation styles, Group, is a shortcut to creating a grouped report. It generates a tabular report that has one group level and some other grouping properties defined. You can then customize the report in the Report painter workspace.

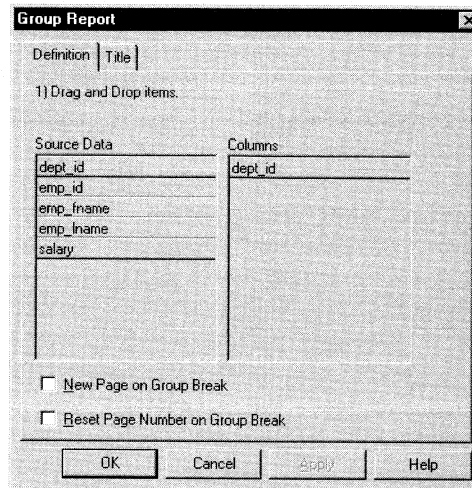
❖ **To create a basic grouped report using the Group presentation style:**

- 1 Click the Report button in the PowerBar.  
The Select Report dialog box displays listing the reports in the current library.
- 2 Click the New button.  
The New Report dialog box displays.
- 3 Choose a data source and the Group presentation style, and click OK.  
You are prompted to define the data for the report.
- 4 Define the data.  
You are prompted to define the grouping column.
- 5 Drag the column(s) you want to group on from the Source Data box to the Columns box.

### Multiple columns and multiple group levels

You can specify more than one column but all columns apply to group level one. You can define one group level at this point. Later in the workspace you can define additional group levels.

In the following example, grouping will be by department, as specified by the dept\_id column:

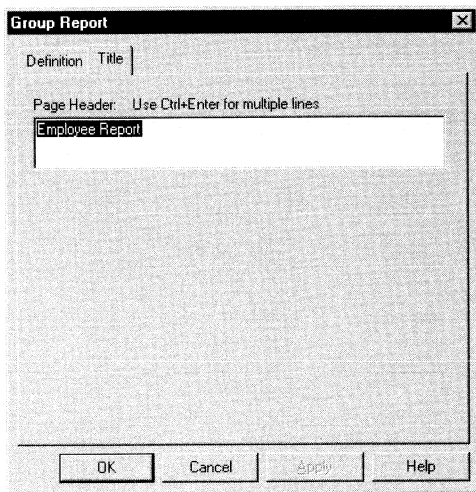


You can double-click a column name and type an expression as the grouping item. You can specify more than one grouping item expression for a group. A break occurs whenever the value concatenated from each column/expression changes.

- 6 If you want a page break each time a grouping value changes, select the New Page On Group Break box.
- 7 If you want page numbering to restart at 1 each time a grouping value changes, select the Reset Page Number On Group Break box.

- 8 Select the Title tab to provide a page header for the report.

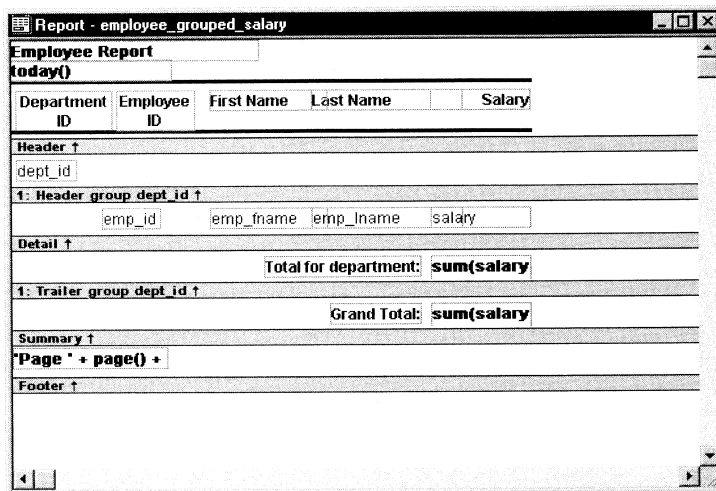
InfoMaker suggests a header based on your data source. For example, if your data comes from the Employee table, InfoMaker uses the name of the table in the suggested header:



- 9 Specify the text.
- 10 Click OK.

The workspace displays with the basic grouping properties set.

This is an example of a Group style report in the workspace:



What InfoMaker does As a result of your specifications, InfoMaker generates a tabular report and:

- ◆ Creates group header and trailer bands
- ◆ Places the column you chose as the grouping column in the group header band
- ◆ Sorts the rows by the grouping column
- ◆ Places the page header and the date (as a computed field) in the header band
- ◆ Places the page number and page count (as computed fields) in the footer band
- ◆ Creates sum computed fields for all numeric columns (the fields are placed in the group trailer and summary bands)

Here is the preceding report during preview:

<b>Employee Report</b>				
<b>8/13/97</b>				
<b>Department ID</b>	<b>Employee ID</b>	<b>First Name</b>	<b>Last Name</b>	<b>Salary</b>
100				
	102	Fran	Whitney	\$45,700
	105	Matthew	Cobb	\$62,000
	160	Robert	Breault	\$57,490
	243	Natasha	Shishov	\$72,995
	247	Kurt	Driscoll	\$48,024
	249	Rodrigo	Guevara	\$42,998
	266	Ram	Gowda	\$59,840
	278	Terry	Melkisetian	\$48,500
	316	Lynn	Pastor	\$74,500

What you can do

You can use any of the techniques available in a tabular report to modify and enhance the grouped report, such as moving objects, specifying display formats, and so on. In particular, see "Defining groups in an existing report" next to learn more about the bands in a grouped report and how to add features especially suited for grouped reports (for example, add a second group level, define additional summary statistics, and so on).

## Defining groups in an existing report

Instead of using the Group presentation style to create a grouped report from scratch, you can take an existing tabular report and define groups in it.

❖ **To add grouping to an existing report:**

- 1 Start with a tabular report that retrieves all the columns you need.
- 2 Specify the grouping columns.
- 3 Sort the rows.
- 4 (Optional) Rearrange the report.
- 5 (Optional) Add summary statistics.
- 6 (Optional) Sort the groups.

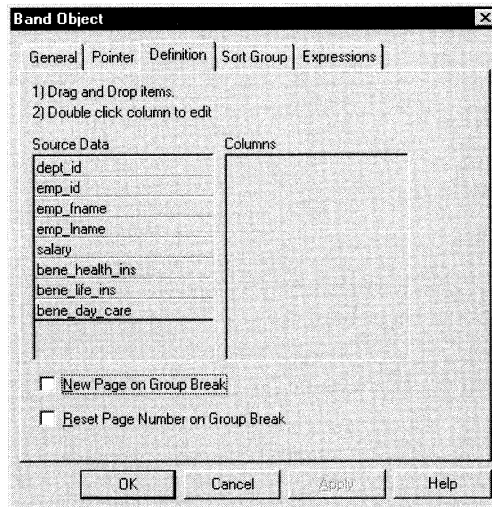
Steps 2 through 6 are described below.

## Specifying the grouping columns

❖ **To specify the grouping columns:**

- 1 In the Report painter, Select Rows>Create Group from the menu bar.

The Band Object property sheet displays:



- 2 Specify the group columns, as described in "Using the Group presentation style" on page 310.

## Creating subgroups

After defining your first group, you can define subgroups, which are groups within the group you just defined.

❖ **To define subgroups:**

- 1 Select Rows>Create Group from the menu bar and specify the column/expression for the subgroup.
- 2 Repeat step 1 to define additional subgroups if you want.

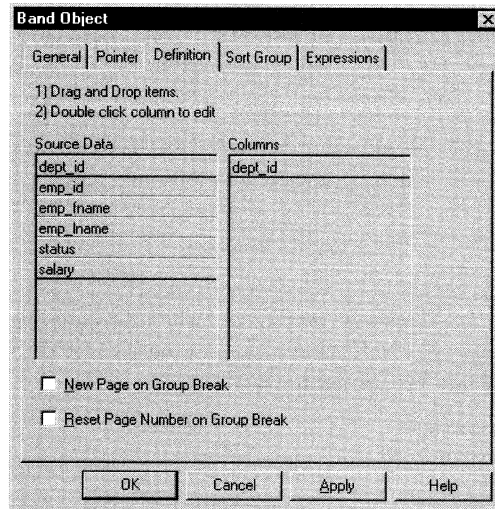
You can specify as many levels of grouping as you need.

## How groups are identified

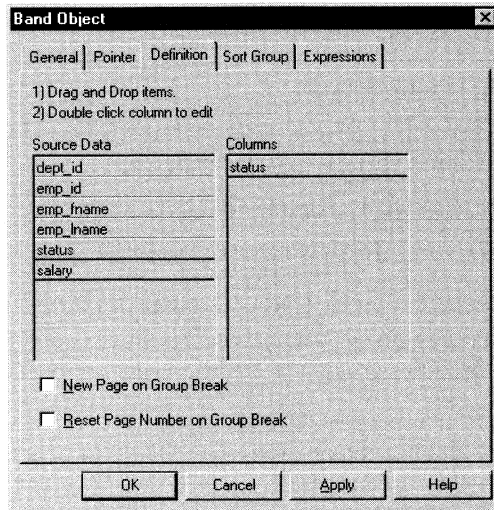
InfoMaker assigns each group a number (or level) when you create the group. The first group you specify becomes group 1, the primary group. The second group becomes group 2, a subgroup within group 1, and so on.

For example, say you defined two groups. The first group uses the dept\_id column and the second group uses the status column.

## First group



## Second group



The rows will be grouped first by department (group 1). Within department, rows will be grouped by status (group 2). If you specify page breaks for the groups, a page break will occur when any of these values changes.

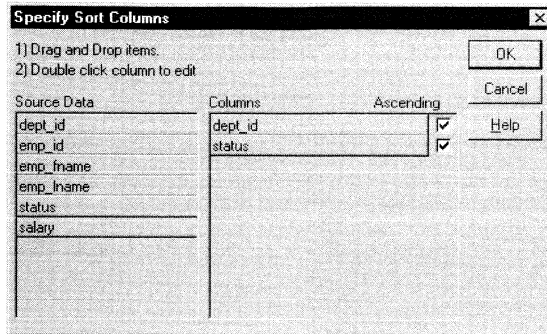
You use the group's number to identify it when defining summary statistics for the group, which is described later in this chapter.

## Sorting the rows

InfoMaker does not sort the data when it creates a group. Therefore, if the data source is not sorted, you must sort the data by the same columns (or expressions) specified for the groups.



For example, if you are grouping by Dept\_ID then Status as shown above, select Rows>Sort from the menu bar and specify Dept\_ID and then Status as sorting columns:



You can also sort on additional rows if you want. For example, if you want to sort by employee ID within each group, specify Emp\_ID as the third sorting column.

For more information about sorting, see "Sorting rows" on page 305.

## Rearranging the report

When you create a group, InfoMaker creates two new bands for each group:

- ◆ A group header band
- ◆ A group trailer band

The bar identifying the band contains:

- ◆ The number of the group
- ◆ The name of the band
- ◆ The name of each column that defines the group
- ◆ An arrow pointing to the band

Department ID	Employee ID	First Name	Last Name	Salary	Health Insurance	Life Insurance
<b>Header</b> ↑						
1: Header group dept_id ↑						
dept_id	emp_id	emp_fname	emp_lname	salary	<input type="checkbox"/> Health Insurance	<input type="checkbox"/> Life Insurance
<b>Detail</b> ↑						
1: Trailer group dept_id ↑						
<b>Summary</b> ↑						
<b>Footer</b> ↑						

You can include any object in the report (such as columns, text, and computed fields) in the header and trailer bands of a group.

Using the group header band

The contents of the group header band display at the top of each page and after each break in the data.

Typically, you use this band to identify each group. For example, you might move the grouping column from the detail band to the group header band, since it now serves to identify one group rather than each row.

For example, if you group the rows by department and include the department in the group header, the department will display before the first line of data each time the department changes.

The screenshot shows a table with columns: Employee ID, First Name, Last Name, Salary, Health Insurance, and Life Insurance. A group header is visible for 'Department ID' with a sub-column 'dept\_id'. Below the header, a '1: Header group dept\_id ↑' row shows the column headers for the data rows. A 'Detail ↑' section follows, with a '1: Trailer group dept\_id ↑' row. A 'Summary ↑' and 'Footer ↑' section are also present but empty.

In preview, you see this:

The screenshot shows the same table with data rows. A group header for 'Department ID' with value '100' is displayed. The data rows are as follows:

Employee ID	First Name	Last Name	Salary	Health Insurance	Life Insurance
862	John	Sheffield	\$87,900.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
102	Fran	Whitney	\$45,700.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
582	Peter	Samuels	\$37,400.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
479	Linda	Siperstein	\$39,875.50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
160	Robert	Breault	\$57,490.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
278	Terry	Melkisetian	\$48,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
249	Rodrigo	Guevara	\$42,998.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
316	Lynn	Pastor	\$74,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
839	Dean	Marshall	\$42,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
243	Natasha	Shishov	\$72,995.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1250	Emilio	Diaz	\$54,900.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Using the group trailer band

The contents of the group trailer display after the last row for each value that causes a break.

In the group trailer band, you specify the information you want displayed after the last line of identical data for each value in the group. Typically, you include summary statistics here, as described next.

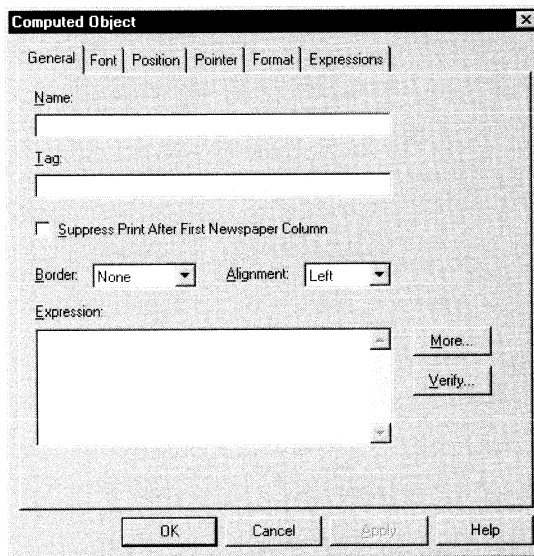
## Adding summary statistics

One of the advantages of creating a grouped report is that you can have InfoMaker calculate statistics for each group. To do that, you place computed fields that reference the group. Typically, you place these computed fields in the group's trailer band.

❖ **To add a summary statistic:**

- 1 Click the Computed Field button in the Objects dropdown toolbar.
- 2 Click the workspace where you want the statistic.

The Computed Object property sheet displays:



The screenshot shows the 'Computed Object' dialog box with the following elements:

- Tabbed interface with 'General' selected.
- 'Name:' text input field.
- 'Tag:' text input field.
- Checkbox:  Suppress Print After First Newspaper Column
- 'Border:' dropdown menu (value: None)
- 'Alignment:' dropdown menu (value: Left)
- 'Expression:' large text area with scrollbars.
- 'More...' and 'Verify...' buttons to the right of the Expression field.
- 'OK', 'Cancel', 'Apply', and 'Help' buttons at the bottom.

- 3 (Optional) Name the computed field (this allows you to reference the computed field later if you need to).
- 4 Specify the expression that defines the computed field (see below).
- 5 Click OK.

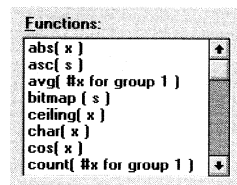
**A shortcut to sum values**

If you want to sum a numeric column, select the column in the workspace and click the Sum button in the Objects dropdown toolbar.

InfoMaker automatically places a computed field in the appropriate band in the workspace.

### Specifying the expression

Typically, you will use aggregate and other functions in your summary statistic. InfoMaker lists functions you can use in the Functions box in the Modify Expression dialog box (to display this dialog box, click More in the Computed Object property sheet). When you are defining a computed field in a group header or trailer band, InfoMaker automatically lists forms of the functions that reference the group:



You can paste these templates into the expression, then replace the #x that is pasted in as the function argument with the appropriate column or expression.

For example, to count the employees in each department (group 1), specify this expression in the group trailer band:

```
Count( Emp_Id for group 1 )
```

To get the average salary of employees in a department, specify:

```
Avg( Salary for group 1 )
```

To get the total salary of employees in a department, specify:

Sum( Salary for group 1 )

Employee ID	First Name	Last Name	Salary
<b>Header ↑</b>			
<b>Department ID</b>			
dept_id			
<b>1: Header group dept_id ↑</b>			
emp_id	emp_fname	emp_lname	salary
<b>Detail ↑</b>			
			<b>Average Salary:</b> avg( salary for
			<b>Total Salary:</b> sum( salary for
<b>1: Trailer group dept_id ↑</b>			
<b>Summary ↑</b>			
<b>Footer ↑</b>			

When you run the report, you see this:

Employee ID	First Name	Last Name	Salary
<b>Department ID</b>			
500			
191	Jeannette	Bertrand	\$29,800
1013	Joseph	Barker	\$27,290
921	Charles	Crowley	\$41,700
868	Felicia	Kuo	\$28,200
1658	Michael	Lynch	\$24,903
1615	Sheila	Romero	\$27,500
750	Jane	Braun	\$34,300
1570	Anthony	Rebeiro	\$34,576
703	Jose	Martinez	\$55,501
			<b>Average Salary:</b> \$33,752
			<b>Total Salary:</b> \$303,770

## Sorting the groups

You can sort the groups in your report. For example, in a report showing employee information grouped by department, you might want to sort the departments (the groups) by total salary.

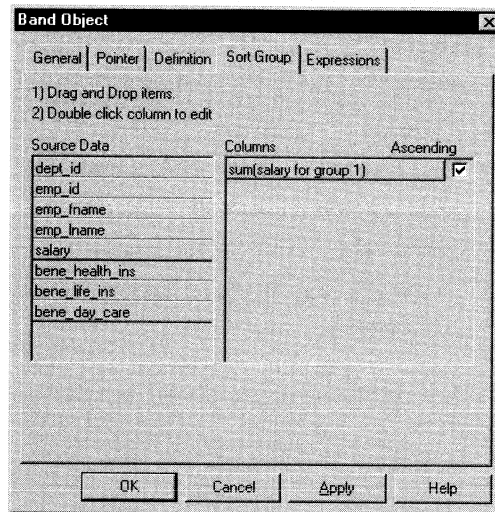
Typically, this involves aggregate functions, as described in "Adding summary statistics" on page 320. In the department salary example, you would sort the groups using the aggregate function Sum to calculate total salary in each department.

❖ **To sort the groups:**

- 1 Place the mouse pointer on the group header bar (not inside the band).  
The pointer becomes a double-headed arrow.
- 2 Display the popup menu for the group header, select Properties, and then select the Sort Group tab.
- 3 Drag the column you want to sort the groups by from the Source Data box into the Columns box.

If you chose a numeric column, InfoMaker uses the Sum function in the expression; if you chose a non-numeric column, InfoMaker uses the Count function.

For example, if you chose the Salary column, InfoMaker specifies that the groups will be sorted by the expression **sum(salary for group 1)**:

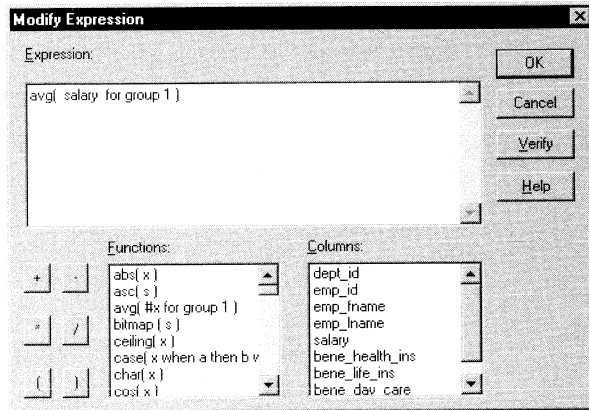


- 4 Select ascending or descending sort as appropriate.
- 5 If you want to modify the expression to sort on, double-click the column in the Columns box.

The Modify Expression dialog box displays.

- Specify the expression to sort on.

For example, to sort the department group (the first group level) on average salary, specify **avg(salary for group 1)**:



- Click OK.

You return to the Band Object property sheet with the expression displayed.

- Click OK.

When you run the report, the groups will be sorted on the expression you specified.



# Highlighting Information in Reports and Forms

## About this chapter

This chapter describes how you modify the way information displays in your reports and forms based on the conditions you specify. The conditions are usually related to data values, which are not available until execution time.

Most of the discussions in this chapter focus on reports. But you can also use the techniques described in this chapter in the Form painter to highlight information in forms.

## Contents

<b>Topic</b>	<b>Page</b>
Overview of highlighting information	326
Conditionally modifying properties at execution time	330
Example 1: creating a gray bar effect	332
Example 2: rotating objects	335
Example 3: highlighting rows of data	337
Example 4: changing the size and location of objects	340
Supplying property values	343
Specifying colors	364

## Overview of highlighting information

### About properties

Every object in your report has a set of *properties* that determines what the object looks like and where it is located. For example, the values in a column of data display in a particular font and color, in a particular location, with or without a border, and so on.

---

### Everything in this chapter applies to both reports and forms

Most of the discussions in this chapter focus on reports and the Report painter. The techniques described also apply to forms and the Form painter.

---

### About modifying properties in the workspace

You define the appearance and behavior of these objects in reports in the Report painter. As you do that, you are specifying the objects' properties. For example, when you place a border around a column in the painter workspace, you are setting that column's Border property.

In most cases, the appearance and behavior of objects is fixed; you do not want them to change at execution time. You make headings bold in the workspace and that's the way you want them to be at all times.

In the following report, the Salary Plus Benefits column has a Shadow box border around every data value in the column. To display the border, you would set the border property on the column's property sheet:

Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$93,177
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$69,650
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$44,892

### About modifying properties at execution time

In some cases, however, you might want some properties of objects in reports to be driven by the data, which is not known when you are defining the report in the painter. For these situations you can define **property conditional expressions**, which are expressions that are evaluated at execution time.

You can use these expressions to conditionally and dynamically modify the appearance and behavior of your report during execution. The results of the expressions set the values of properties of objects in the report.

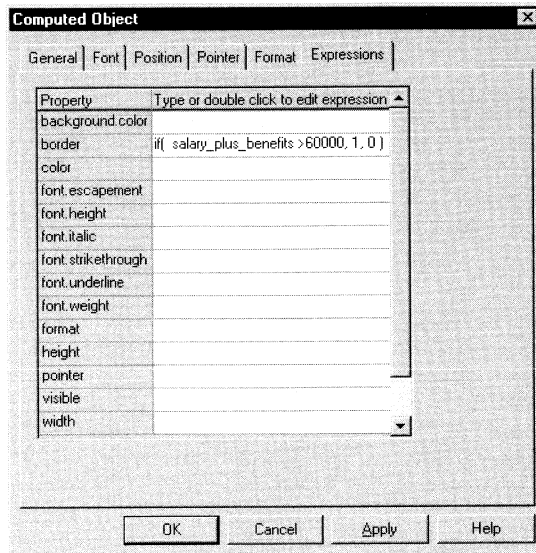
In the following report, the Salary Plus Benefits column has a Shadow box border highlighting each data value that is greater than \$60,000:

Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$93,177
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$69,650
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$44,892

To control the display of the border, you would enter a property conditional expression on the Expressions property page of the column's property sheet. When you run the report, InfoMaker changes the border of individual data values based on your expressed condition (value greater than \$60,000).

About filling in the Expressions property page

The following Expressions property page lists properties for the Salary Plus Benefits column, which is a computed object. In this example, you want to change the Border property, so you put the conditional expression next to Border:



A closer look at the expression

The expression you enter almost always begins with If. Then you specify three things: the condition, what happens if it's true, and what happens if it's false. Parentheses surround the three things and commas separate them:

```
If( expression, true, false )
```

This is the expression used in the example. Because the expression is next to the Border property, the values for true and false indicate particular borders. The value 1 means Shadow box border and the value 0 means no border:

```
If(salary_plus_benefits > 60000, 1, 0)
```

When you run the report, InfoMaker checks the value in the computed column called salary\_plus\_benefits to see if it is greater than 60000. If it is (true), InfoMaker will display the value with the Shadow box border. If not (false), InfoMaker will display the value with no border.

About specifying properties

Usually you specify a number to indicate what you want for a particular property. For example, the following list shows all of the borders you can specify and the numbers you use. If you want the border property to be Shadow box, you specify **1** in the If statement, for either true or false.

**0**—None

- 1—Shadow box
- 2—Box
- 3—Resize
- 4—Underline
- 5—3D Lowered
- 6—3D Raised

FOR INFO For details on the values of properties that can be set on the Expressions property page, see "Supplying property values" on page 343.

## Conditionally modifying properties at execution time

Conditionally modifying properties at execution time is a technique you can use to highlight information in a form or a report.

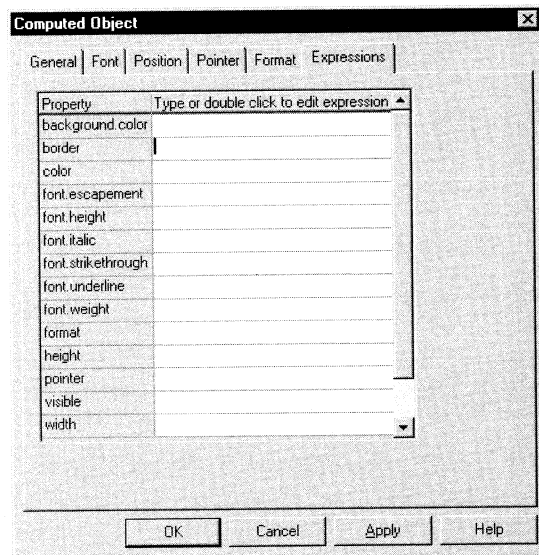
FOR INFO For an explanation of this technique, see "Overview of highlighting information" on page 326.

### ❖ To conditionally modify properties at execution time:

- 1 Position the pointer on the object whose properties you want to modify during execution.
- 2 Select Properties from the object's popup menu and then select the Expressions property page.
- 3 Enter the appropriate expression next to the property you want to change.

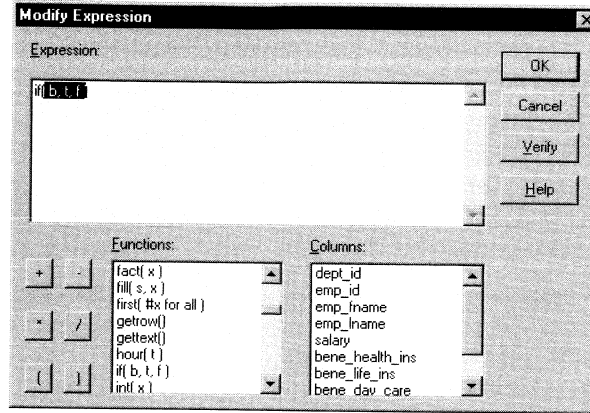
An easy way to enter or change an expression is to use the Modify Expression dialog box.

- 4 To access the Modify Expression dialog box, double-click the line containing the property for which you want to enter an expression:



The Modify Expression dialog box displays.

- 5 Scroll the list of functions in the Functions box until you see the IF function and then select it:



- 6 Replace the *b* (boolean) with your condition (for example, `salary>40000`).

You can select columns and functions and use the buttons to add the symbols shown on them.

Replace the *t* (true) with the value to use for the property if the condition is true.

Replace the *f* (false) with the value to use for the property if the condition is false.

Values to use for properties are usually numbers. They are different for each property.

**FOR INFO** For more information about property values that can be set on the Expressions page, see "Supplying property values" on page 343.

- 7 Click OK to return to the Expressions property page.
- 8 If you want, enter expressions for other properties of the object.
- 9 Click OK.

You return to the Report painter workspace.

**FOR INFO** For examples, see "Example 1: creating a gray bar effect" next, "Example 2: rotating objects" on page 335, "Example 3: highlighting rows of data" on page 337, and "Example 4: changing the size and location of objects" on page 340.

## Example 1: creating a gray bar effect

The following report shows alternate rows with a light gray bar. The gray bars make it easier to track data values across the row:

Total Compensation Report Salary Plus Benefits								Page 1 of 4 01/10/98
<i>Value of life insurance = \$(5.43 × salary)/1,000</i> <i>Value of day care = \$5,200</i>								
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
100	102	Fran	Whitney	\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
	105	Matthew	Cobb	\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
	160	Robert	Breault	\$57,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
	243	Natasha	Shishov	\$72,995	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
	247	Kurt	Driscoll	\$48,024	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
	249	Rodrigo	Guevara	\$42,998	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
	266	Ram	Gowda	\$59,840	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
	278	Terry	Melkisetian	\$48,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
	316	Lynn	Pastor	\$74,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905

Adding a rectangle object

To create the gray bar effect, you would add a rectangle object to the detail band and size it so that it surrounds the objects you want highlighted. The rectangle needs to be located in the band (Rectangle property sheet>Position tab>Layer option>Band setting). The rectangle also needs to be specified as Send to Back (select from the popup menu) so that it will be behind all the values:

1: Header_group_dept_id ↑								
dept_id	emp_id	emp_fname	emp_lname	salary	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	salary + if
Detail ↓								

At this point, the report shows a rectangle for each detail line. A narrow black line bounds the rectangle:

Total Compensation Report Salary Plus Benefits								Page 1 of 4 01/10/98
<i>Value of life insurance = \$(5.43 × salary)/1,000</i> <i>Value of day care = \$5,200</i>								
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
100	102	Fran	Whitney	\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
	105	Matthew	Cobb	\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
	160	Robert	Breault	\$57,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
	243	Natasha	Shishov	\$72,995	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
	247	Kurt	Driscoll	\$48,024	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
	249	Rodrigo	Guevara	\$42,998	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
	266	Ram	Gowda	\$59,840	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
	278	Terry	Melkisetian	\$48,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
	316	Lynn	Pastor	\$74,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905



The line bounding the rectangle should be set to None (Rectangle property sheet>General tab>Line option>None setting). You would do this last, after positioning the rectangle object.

At this point the report looks as it did originally without the rectangles in the detail lines. The rectangles are there but because they are white and are not bounded with a black line, you cannot see them.

#### Specifying the condition

Now you would specify the condition for making the rectangle gray. To do this, you would select Properties from the popup menu for the rectangle. On the Expressions property page, you would enter the following expression for the Brush.Color property:

```
If(mod(getrow(),2)=1, rgb(255, 255, 255 ), rgb(240, 240, 240 ))
```

The expression  $\text{mod}(\text{getrow}(),2)=1$  distinguishes odd rows from even rows.

The function  $\text{mod}(\text{getrow}(),2)$  takes the row number ( $\text{getrow}()$ ), divides it by 2, then returns the remainder. The remainder can be either 0 or 1. If the row number is odd, mod returns 1; if the row number is even, mod returns 0.

If the row number is odd (the condition evaluates as true), the rectangle displays as white. The rgb function specifies maximum amounts of red, green, and blue:  $\text{rgb}(255, 255, 255)$ . Specifying 255 for red, green, and blue results in the color white. If the row number is even (the condition evaluates as false), the rectangle displays as light gray ( $\text{rgb}(240, 240, 240)$ ).

Finishing up

Now the report has gray bands. The final adjustment is to change the background color of the computed field used for Salary Plus Benefits. In the workspace, change the background color of the computed field from white to transparent (Computed object property sheet>Font tab>Background option>Transparent):

Total Compensation Report		Page 1 of 4						
Salary Plus Benefits		01/10/98						
		Value of life insurance = \$(5.43 x salary)/1,000						
		Value of day care = \$5,200						
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
100	102	Fran	Whitney	\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
	105	Matthew	Cobb	\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
	160	Robert	Breault	\$57,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
	243	Natasha	Shishov	\$72,995	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
	247	Kurt	Driscoll	\$48,024	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
	248	Rodrigo	Guevara	\$42,998	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
	266	Ram	Gowda	\$59,840	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
	278	Terry	Melkisetian	\$48,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
	316	Lynn	Pastor	\$74,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905

## Example 2: rotating objects

The following report shows column headers rotated 45 degrees.

### On Macintosh rotation is limited

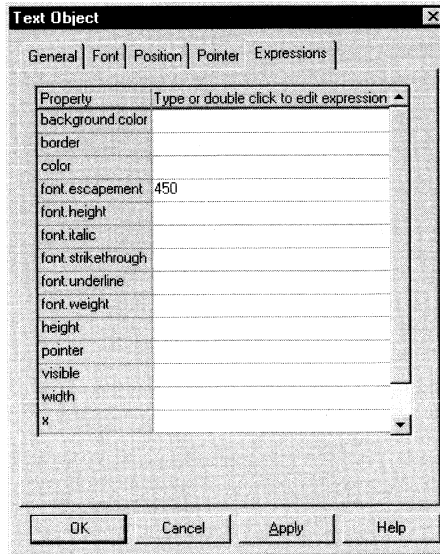
On the Macintosh, you can rotate a text object 90 degrees only. Other values are ignored.

Total Compensation Report					Page 1 of 4			
Salary Plus Benefits					01/10/98			
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
100	102	Fran	Whitney	\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
	105	Matthew	Cobb	\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
	160	Robert	Breault	\$57,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
	243	Natasha	Shishov	\$72,995	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
	247	Kurt	Driscoll	\$48,024	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
	249	Rodrigo	Guevara	\$42,998	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
	266	Ram	Gowda	\$58,840	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
	278	Terry	Melkisetian	\$48,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
	316	Lynn	Pastor	\$74,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905

To rotate each of these three text objects, you would select Properties from the popup menu for each object and make an entry on the Expressions property page.

Specifying rotation

To specify rotation, you would enter the number 450 next to the Font.Escapement property. The number 450 means 45 degrees; the value entered for Font.Escapement is in tenths of degrees:



Notice that you do not have to specify a condition. Typically, you would not specify a condition for object rotation. When you have specified 450 for the three text objects, you end up with all three objects rotated:

Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
\$57,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802

If you try rotation, you may notice that the objects do not appear to be rotated in the workspace. You see the rotation only when you run the report.

## Example 3: highlighting rows of data

The following report is an employee phone list. Out-of-state (not in Massachusetts) employees are shown in bold and preceded by two asterisks (\*\*):

<i>(** means out of state)</i>		Employee Phone List	
		01/10/98	
Employee Name	Phone	Employee Name	Phone
Ahmed, Alex	(617) 555-8748	** Overbey, Rollin	<b>(510) 555-7255</b>
Barker, Joseph	(617) 555-8021	Pastor, Lynn	(617) 555-2001
Barletta, Irene	(617) 555-8345	Pickett, Catherine	(617) 555-3478
Bertrand, Jeannette	(508) 555-8138	Poitras, Kathleen	(617) 555-3920
Bigelow, Janet	(617) 555-1493	Powell, Thomas	(617) 555-1956
Blaikie, Barbara	(617) 555-9345	Preston, Mark	(617) 555-5862
Braun, Jane	(617) 555-7857	Rabkin, Andrew	(617) 555-4444
Breault, Robert	(617) 555-3099	Rebeiro, Anthony	(617) 555-5737
Bucceri, Matthew	(617) 555-5336	Romero, Sheila	(617) 555-8138
Butterfield, Joyce	(617) 555-2232	Samuels, Peter	(617) 555-8342
Chao, Shih Lin	(617) 555-5921	** Savarino, Pamela	<b>(310) 555-1857</b>
Charlton, Doug	(508) 555-9246	Scott, David	(617) 555-3246
** Chin, Philip	<b>(404) 555-2341</b>	Shea, Mary Anne	(617) 555-4616
** Clark, Alison	<b>(510) 555-9437</b>	** Sheffield, John	<b>(713) 555-3877</b>
Cobb, Matthew	(617) 555-3840	Shishow, Natasha	(617) 555-2755
Coe, Kristen	(617) 555-9192	Singer, Samuel	(508) 555-3255
Coleman, James	(508) 555-4735	Siperstein, Linda	(617) 555-6588
Crow, John	(617) 555-3332	** Sisson, Thomas	<b>(713) 555-8390</b>
Crowley, Charles	(617) 555-9425	** Smith, Susan	<b>(713) 555-6613</b>
Davidson, Jo Ann	(617) 555-3870	Soo, Hing	(617) 555-8748
Diaz, Emilio	(617) 555-3567	Sterling, Paul	(508) 555-0295

FOR INFO This report uses newspaper columns. To understand how to create this report without highlighting data, see "Printing with newspaper-style columns" on page 225.

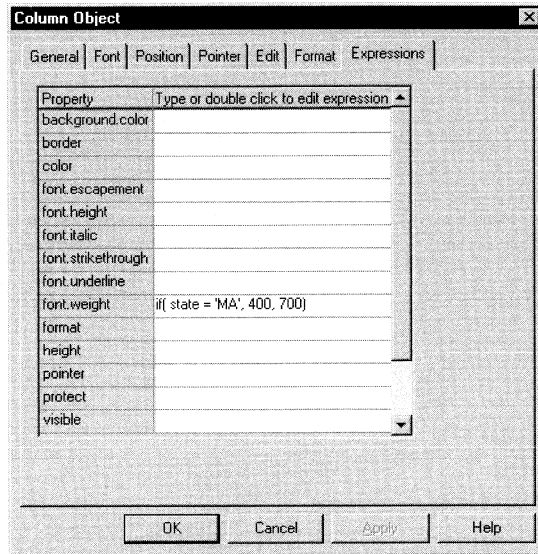
In the workspace, the detail band looks like this. It includes four objects: the employee last name, a comma, the employee first name, and the phone number:

Header ↑
emp_lname   emp_fname   phone
Detail ↑

Making the objects display in bold

To make these objects display in bold if the employee is not from Massachusetts, you would use the Expressions property page. To do this, select Properties from the popup menu of each object. For each object, enter the following expression next to the Font.Weight property:

```
If(state = 'MA', 400, 700)
```



The expression states: If the value of the state column is MA (if this is true), use 400 as the font weight. This means employees from Massachusetts will display in the normal font. For false (any state but MA), use 700 as the font weight. This means all other employees display in bold font.

---

### Logic that relies on the state column

To use logic that relies on the state column, you need to include it in the data source. You can add the column after creating the report by modifying the data source. Also notice that the state column does not actually appear anywhere in the report. Values must be available but do not need to be included in the report.

---

Adding the two asterisks (\*\*)

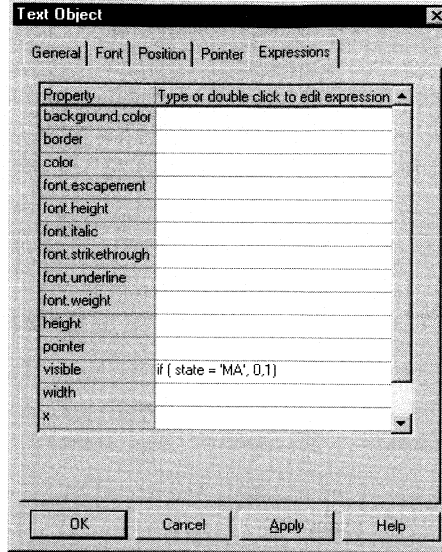
To insert two asterisks (\*\*) in front of the employee name if the employee is not from Massachusetts, you would first add a text object with the two asterisks in bold, exactly as you want them to display:

The image shows a report table with a header row and a detail row. The header row contains the text '\*\*emp\_fname | emp\_fname | phone'. The detail row contains the text 'emp\_fname | emp\_fname | phone'. The two asterisks in the header row are bolded.

Header ↑
**emp_fname   emp_fname   phone
Detail ↑

Then you use logic on the Visible property. To do this, display the popup menu and select Properties. On the Expressions property page, enter the following expression next to the Visible property:

```
If(state = 'MA', 0, 1)
```



This expression says: If the state of the employee is MA (the true condition), the Visible property of the \*\* object is off (indicated by 0). If the state of the employee is not MA (the false condition), the Visible property of the \*\* object is on (indicated by 1). The asterisks will be visible next to that employee's name.

---

**Tip**

You can use underlines, italics, strikethrough, borders, and colors to highlight your information. Experiment to get the effects you like.

---

## Example 4: changing the size and location of objects

The following report shows a rectangle and a line that change size and location if the current row contains data for a customer from the state of New York. The City and State columns also change location, but the discussion will focus on the rectangle and the line. The process for the columns is analogous:

Customer ID	Name	Address	City	State
101	Michaels Devlin	3114 Pioneer Avenue	Rutherford	NJ
102	Beth Reiser	1033 Whippany Road		New York NY
103	Erin Niedringhaus	1990 Windsor Street	Panoli	PA
104	Meghan Mason	550 Dundas Street East	Knowville	TN
105	Laura McCarthy	1210 Highway 36	Carmel	IN

In the workspace, the rectangle and line display in one location (and with a single set of dimensions):

Customer ID	Name	Address	City	State
101	Michaels Devlin	3114 Pioneer Avenue	Rutherford	NJ
102	Beth Reiser	1033 Whippany Road		New York NY
103	Erin Niedringhaus	1990 Windsor Street	Panoli	PA
104	Meghan Mason	550 Dundas Street East	Knowville	TN
105	Laura McCarthy	1210 Highway 36	Carmel	IN

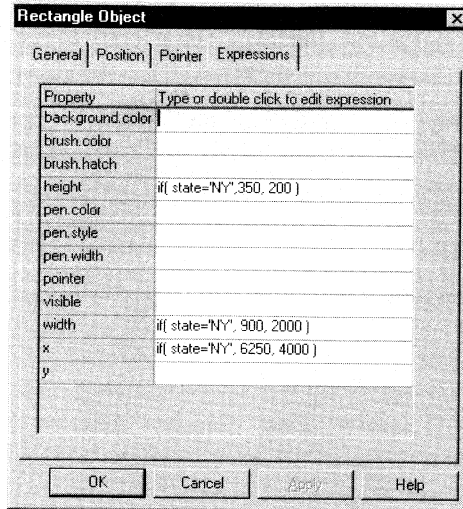
To change properties of the rectangle and the line for rows with the state column equal to New York, you would specify the following If statement for every size or location property you wanted to change. You would substitute the values appropriate to the property. All of the values used in this example are in thousandths of inches, the unit of measure used for the report:

`If(state='NY', true value, false value)`

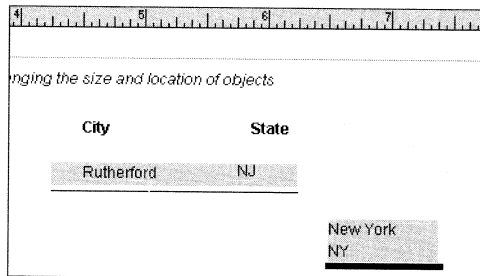


Changing properties  
of the rectangle

For the rectangle, the Expressions property page has the entries shown for the width and height of the rectangle and for its x coordinate:



At execution time, the rectangle is one size and x location if the state is NY, and another size and location if the state is not NY. You may notice that the x values seem to be off by .25. This is because the left margin is set to .25, which adds .25 to each x value:

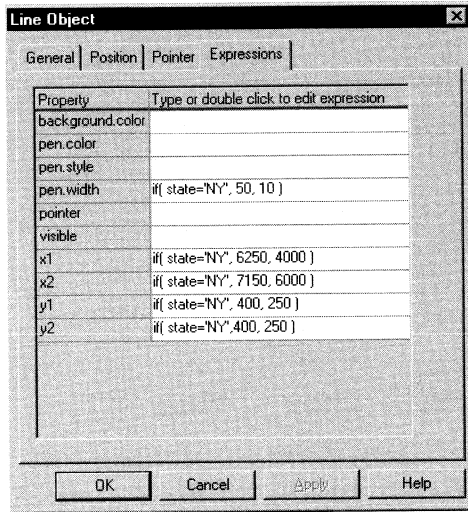


#### Example 4: changing the size and location of objects

---

Changing properties of the line

For the line, the Expressions property page has entries for two sets of x1 and x2 coordinates of the line, which define two different line lengths and x positions. It also has y1 and y2 coordinates, which define the two different y positions of the line. Finally, it has an entry for pen width, which defines two different pen widths:



At execution time, the line changes length, x and y position, and pen width depending on whether the state for the current row of information is NY.

## Supplying property values

Each property has its own set of property values that you can use to specify for the true and false conditions in your If expression. Usually you specify a number to indicate what you want. For example, if you are working with the Border property, you use the number 0, 1, 2, 3, 4, 5, or 6 to specify a border.

This section describes the valid property values for all of the properties you can control in the Expressions property page. These are the properties:

<b>Property</b>	<b>What you can specify</b>
Background.Color	Background color of an object
Border	Border of an object
Brush.Color	Color of a graphic object
Brush.Hatch	Pattern used to fill a graphic object
Color	Color of text for text objects, columns, and computed fields
Font.Escapement (for rotating objects)	Rotation of an object
Font.Height	Height of text
Font.Italic	Use of italic font for text
Font.Strikethrough	Use of strikethrough for text
Font.Underline	Use of underlining for text
Font.Weight	Weight (for example, bold) of text font
Format	Display format for columns and computed fields
Height	Height of an object
Pen.Color	Color of a line or the line surrounding a graphic object
Pen.Style	Style of a line or the line surrounding a graphic object
Pen.Width	Width of a line or the line surrounding a graphic object
Pointer	Image to be used for the pointer
Visible	Whether an object is visible
Width	Width of an object
X	X position of an object

Property	What you can specify
X1, X2	X coordinates of either end of a line
Y	Y position of an object relative to the band in which it is located
Y1, Y2	Y coordinates of either end of a line

## Background.Color

**Description** Setting for the background color of an object.

**Value** A number that specifies the object's background color.

**FOR INFO** For information on specifying colors, see "Specifying colors" on page 364.

---

**Background color of a line** The background color of a line is the color that displays between the segments of the line when the pen style is not solid.

**Transparent background** If Background.Mode is transparent (1), Background.Color is ignored.

---

**Example** The following statement specifies that if the current row (person) uses the day care benefit, the background color of the object should be set to light gray (15790320). If not, the background color should be set to white (16777215):

```
If (bene_day_care = 'Y', 15790320, 16777215)
```

In this example, the condition is applied to the Background.Color property in the Expressions property page for three objects: the emp\_id column, the emp\_fname column, and the emp\_lname column.

The following is a portion of the resulting report. Notice that the employee ID, first name, and last name have a gray background if the employee uses the day care benefit:

Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care	Salary Plus Benefits
102	Fran	Whitney	\$45,700	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$50,748
105	Matthew	Cobb	\$62,000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$67,137
160	Robert	Breault	\$57,490	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$67,802
243	Natasha	Shishov	\$72,995	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$78,191
247	Kurt	Driscoll	\$48,024	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,284
249	Rodrigo	Guevara	\$42,998	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$48,031
266	Ram	Gowda	\$59,840	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$60,165
278	Terry	Melkisetian	\$48,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$58,763
316	Lynn	Pastor	\$74,500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	\$84,905
445	Kim	Lull	\$87,900	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$93,177

## Border

Description

The type of border for the object.

Value

A number that specifies the type of border. Values are:

- 0—None
- 1—Shadow box
- 2—Box
- 3—Resize
- 4—Underline
- 5—3D Lowered
- 6—3D Raised

Example

The following statement specifies that if the current row (person) has a status of L (is on leave), the status column should display with a Shadow box border:

```
If(status = 'L', 1, 0)
```

In this example, the condition is applied to the Border property of the status column.

The following is a portion of the resulting report. Notice that the status On Leave displays with a Shadow box border:

Emp ID	Employee First Name	Employee Last Name	Street	City	State	Zip Code	Phone	Status	Sec. No.
102	Fran	Wibbey	49 East Washington Street	Needham	MA	02192-	617) 555-3985	Active	017-34-9033
105	Mathew	Cobb	77 Pleasant Street	Waltham	MA	02154-	617) 555-3640	Active	052-34-5739
129	Philip	Clin	59 Pond Street	Atlanta	GA	30339-	404) 555-2341	Active	02440-8923
148	Julie	Jordan	144 Great Plain Avenue	Winchester	MA	01890-	617) 555-7835	Active	501-70-4733
160	Robert	Breault	58 Cherry Street	Milton	MA	02186-	617) 555-3098	Active	025-48-7623
184	Melissa	Espinosa	112 Ankle Tree Way	Stow	MA	01775-	608) 555-2319	Active	025-48-1943
191	Jeanette	Bertrand	209 Concord Street	Acton	MA	01720-	608) 555-8138	Active	017-34-8821
195	Marv	Dill	89 Hancock Street	Milton	MA	02166-	617) 555-2144	Active	079-48-6634
207	Jane	Francis	12 Handicome Drive	Concord	MA	01742-	608) 555-9022	Active	501-70-8992
243	Natasha	Shishov	15 Milk Street	Waltham	MA	02154-	617) 555-2755	Active	043-21-6799
247	Kurt	Driscoll	154 School Street	Waltham	MA	02154-	617) 555-1234	On Leave	02440-1768
249	Roshoo	Guevara	East Main Street	Framingham	MA	01701-	608) 555-0029	Active	084-32-9990
266	Ram	Gowda	79 Page Street	Natick	MA	01740-	608) 555-6722	Active	017-34-6122

### About the value L and the value On Leave

The status column uses an edit style. The internal value for on leave is L and the display value is On Leave. The conditional expression references the internal value L, which is the actual value stored in the database. The report shows the value On Leave, which is the display value assigned to the value L in the code table for the Status edit style.

## Brush.Color

Description

Setting for the fill color of a graphic object.

Value

A number that specifies the color that fills the object.

FOR INFO For information on specifying colors, see "Specifying colors" on page 364.

Example

See the example for Brush.Hatch, next.

## Brush.Hatch

Description

Setting for the fill pattern of a graphic object.

Value

A number that specifies the pattern that fills the object. Values are:

0—Horizontal

1—Bdiagonal (lines from lower left to upper right)

- 2—Vertical
- 3—Cross
- 4—Fdiagonal (lines from upper left to lower right)
- 5—DiagCross
- 6—Solid
- 7—Transparent

**Example**

In this example, statements check the employee's start date to see if the month is the current month or the month following the current month. Properties of a rectangle object placed behind the row of data are changed to highlight employees with months of hire that match the current month or the month following the current month.

The workspace includes columns of data and a rectangle behind the data. The rectangle has been changed to black in the following picture to make it stand out:

today() Performance Review Reminder				
Department ID	Employee ID	Employee First Name	Employee Last Name	Start Date
Header ↑				
dept_id	emp_id	emp_fname	emp_lname	start_date
Detail ↑				

The following statement is for the `Brush.Color` property of the rectangle. If the month of the start date matches the current month or the next one, `Brush.Color` is set to light gray (12632256). If not, it's set to white (16777215), which means it will not show:

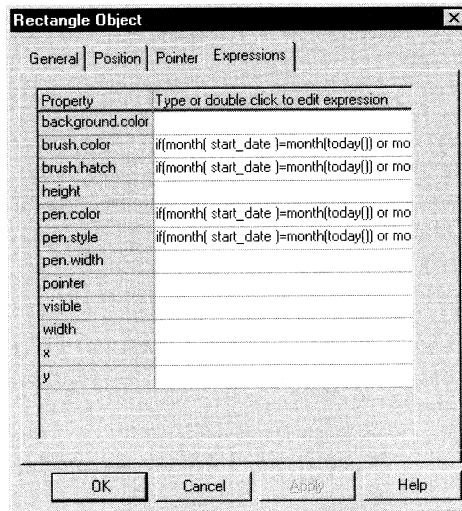
```
If(month( start_date ) = month(today())
or month( start_date ) = month(today())+1
or (month(today()) = 12 and month(start_date)=1),
12632256, 16777215)
```

The following statement is for the `Brush.Hatch` property of the rectangle. If the month of the start date matches the current month or the next one, `Brush.Hatch` is set to `Bdiagonal (1)`. If not, it's set to `Transparent (7)`, which means it will not show:

```
If(month( start_date ) = month(today())
or month( start_date ) = month(today())+1
or (month(today()) = 12 and month(start_date)=1),
1, 7)
```

The Expressions property page looks like this. Notice that it also includes expressions for `Pen.Color` and `Pen.Style`.

FOR INFO For more about these properties, see "Pen.Style" on page 358.



The following is a portion of the resulting report. A rectangle with light gray cross-hatching highlights employees whose reviews are due soon:

01/11/98		Performance Review Reminder		
Department ID	Employee ID	Employee First Name	Employee Last Name	Start Date
400	888	Doug	Charlton	03/10/1991
200	902	Maira	Kelly	04/01/1991
200	913	Ken	Martel	04/16/1991
500	921	Charles	Crowley	04/22/1991
200	930	Ann	Taylor	05/08/1991
200	949	Pamela	Savarino	05/08/1991
100	958	Thomas	Sisson	07/16/1991
400	992	Joyce	Butterfield	08/13/1991
500	1013	Joseph	Barker	09/10/1991
200	1021	Paul	Sterling	10/28/1991
200	1039	Shih Lin	Chao	11/11/1991
400	1062	Barbara	Blaikie	11/20/1991
100	1090	Susan	Smith	12/13/1991
200	1101	Mark	Preston	01/09/1992
200	1142	Alison	Clark	01/19/1992
100	1157	Hing	Soo	01/29/1992
200	1162	Kevin	Goggin	02/03/1992
400	1191	Matthew	Bucceri	02/12/1992



## Color

**Description** The text color of the column, computed field, or text object.

**Value** A number that specifies the color used for text.

**FOR INFO** For information on specifying colors, see "Specifying colors" on page 364.

**Example** The following statement is for the Color property of the emp\_id, emp\_fname, emp\_lname, and emp\_birth\_date columns.

If the employee has a birthday in the current month, the information for the employee displays in red (255). Otherwise, the information displays in black (0):

```
If(month(birth_date) = month (today()), 255, 0)
```

The Expressions property page also includes the same conditional expression for the Font.Underline property so that the example shows clearly on paper.

The following shows a portion of the report. Employees with birthdays in the current month display underlined and in red:

Employee ID	Employee First Name	Employee Last Name	Birth Date
102	Fran	Whitney	06/05/1958
105	Matthew	Cobb	12/04/1960
129	Philip	Chin	10/30/1966
148	Julie	Jordan	12/13/1951
160	Robert	Breault	05/13/1947
184	Melissa	Espinoza	12/14/1939
191	Jeannette	Bertrand	12/21/1964
195	Marc	Dill	07/19/1963
<u>207</u>	<u>Jane</u>	<u>Francis</u>	<u>09/12/1954</u>
243	Natasha	Shishov	04/22/1949
247	Kurt	Driscoll	03/05/1955
249	Rodrigo	Guevara	11/23/1956

## Font.Escapement (for rotating objects)

**Description** The angle of rotation from the baseline of the text.

**Value** An integer in tenths of degrees. For example, 450 means 45 degrees. 0 is horizontal.

---

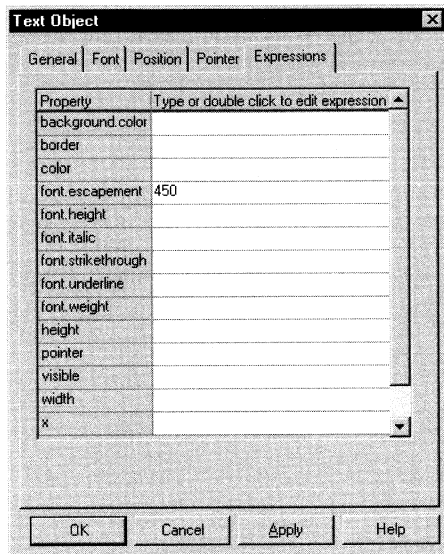
**On Macintosh rotation is limited**

On the Macintosh, you can specify 90 degrees of rotation only. Other values are ignored.

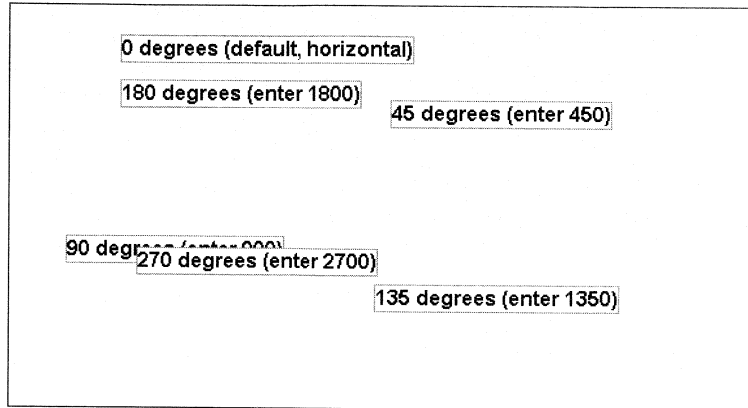
---

**Example**

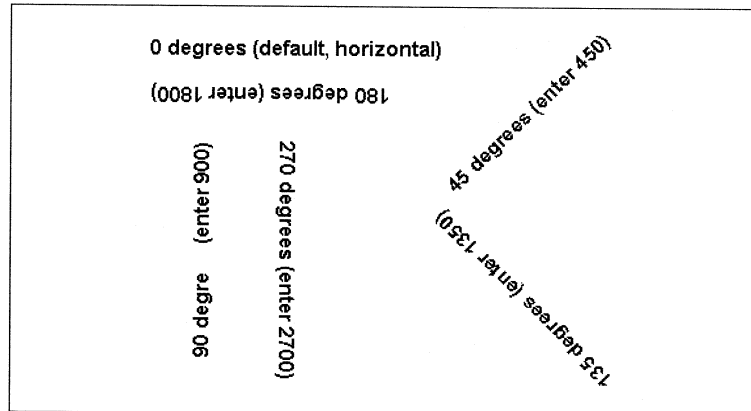
The following Expressions property page shows how you enter rotation for an object. You do not need a conditional expression. Usually you just enter the number of tenths of degrees you want to rotate an object next to the Font.Escapement property:



The following picture shows the workspace with a number of text objects. Each text object shows the Font.Escapement value entered and the number of degrees of rotation. In the workspace, you do not see rotation; it looks as if the objects are all mixed up. Two objects seem to overlay each other:



The next picture shows the same objects at execution time. Each object is rotated appropriately:



---

### How to position objects that are rotated

In the workspace, make the objects movable (display each object's property sheet and select the Moveable checkbox in the Position property page). Then in preview, drag the rotated objects where you want them. To drag a rotated object, you position the pointer in the center of the object and drag from there. When you go back to the workspace, the objects will be wherever you dragged them. They may look incorrectly positioned in the workspace, but they will be correctly positioned when you run the report. When you are satisfied with the positioning, you may want to clear the Moveable checkbox for the objects (to ensure that they stay where you want them).

---

## Font.Height

Description

The height of the text.

Value

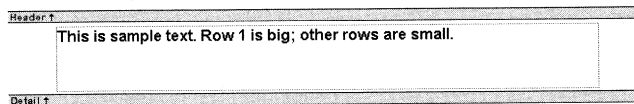
An integer in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels. To specify size in points, specify a negative number.

Example

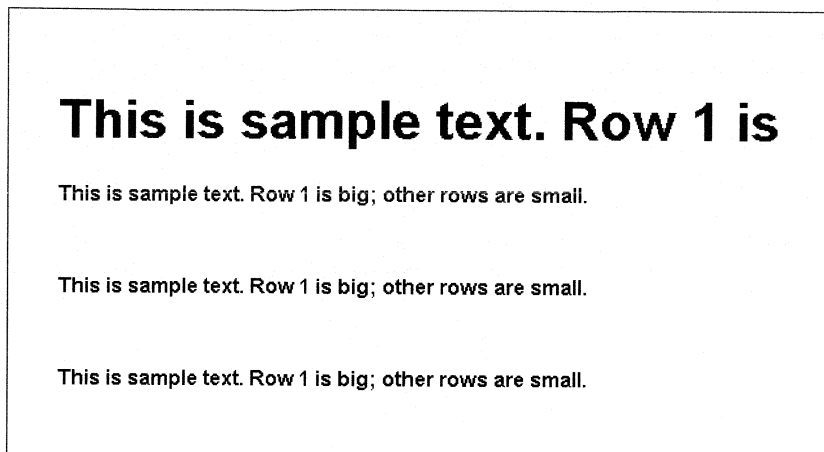
The following statement is included on the Expressions property page for the Font.Height property of a text object. The statement says if it's the first row, show the text 1/2-inch high (500 1/1000ths of an inch) and if it's not the first, show the text 1/5-inch high (200 1/1000ths of an inch):

```
If(GetRow() = 1, 500, 200)
```

The following picture shows the text object in the workspace. Notice that the boundaries of the object had to be extended to allow for the increased size of the text:



This is what the report looks like at execution time. The first occurrence of the text object is big (1/2 inch); subsequent ones are small (1/5 inch). Note that the report is defined as using thousandths of an inch as its unit of measure:



## Font.Italic

Description

A number that specifies whether the text should be italic.

Value

Values are:

**0**—Not italic

**1**—Italic

Example

The following statements are included on the Expressions property page for the Font.Italic, Font.Underline, and Font.Weight properties, respectively. If the employee has health insurance, the employee's information displays in italics. If not, the employee's information displays in bold and underlined:

```
If(bene_health_ins = 'Y', 1, 0)
If(bene_health_ins = 'N', 1, 0)
If(bene_health_ins = 'N', 700, 400)
```

The Expressions property page is filled out in this way for four objects in the workspace: the emp\_id column, the emp\_fname column, the emp\_lname column, and the emp\_salary column.

The resulting report looks like this. Those with health insurance display in italics. Those without health insurance are emphasized with bold and underlining:

				Health insurance		
129	<i>Philip</i>	<i>Chin</i>	<i>\$38,500.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
195	<i>Marc</i>	<i>Dill</i>	<i>\$54,800.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
299	<i>Rollin</i>	<i>Overbey</i>	<i>\$39,300.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
467	<i>James</i>	<i>Klobucher</i>	<i>\$49,500.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
641	<i>Thomas</i>	<i>Powell</i>	<i>\$54,600.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>667</b>	<b>Mary</b>	<b>Garcia</b>	<b>\$39,800.00</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>690</b>	<b>Kathleen</b>	<b>Poitras</b>	<b>\$46,200.00</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
856	<i>Samuel</i>	<i>Singer</i>	<i>\$34,892.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
902	<i>Maira</i>	<i>Kelly</i>	<i>\$87,500.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>913</b>	<b>Ken</b>	<b>Martel</b>	<b>\$55,700.00</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
930	<i>Ann</i>	<i>Taylor</i>	<i>\$46,890.00</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## Font.Strikethrough

**Description** A number that specifies whether the text should be crossed out.

**Value** Values are:

**0**—Not crossed out

**1**—Crossed out

**Example** The following statement is for the Font.Strikethrough property of the emp\_id, emp\_fname, emp\_lname, and emp\_salary columns. The status column must be included in the data source even though it does not appear in the report itself. The statement says that if the employee's status is L, which means On Leave, cross out the text in the object:

```
If(status = 'L', 1, 0)
```

The workspace includes four objects (emp\_id, emp\_fname, emp\_lname, and emp\_salary columns) for which the Expressions property page includes the entry for the Strikethrough property. An extra text object is included to the right of the detail line. It only becomes visible if the status of the row is L (see "Visible" on page 360).

The following is a portion of the resulting report. It shows two employees who are On Leave. The four columns of information show as crossed out:

102	Fran	Whitney	\$45,700.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
105	Matthew	Cobb	\$62,000.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
160	Robert	Breault	\$57,490.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
243	Natasha	Shishov	\$72,995.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
247	<del>Kurt</del>	<del>Driscoll</del>	<del>\$48,023.69</del>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<i>On leave</i>
249	Rodrigo	Guevara	\$42,998.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
266	Ram	Gowda	\$59,840.00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
278	Terry	Melkisetian	\$48,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
316	Lynn	Pastor	\$74,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
445	Kim	Lull	\$87,900.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
453	Andrew	Rabkin	\$64,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
479	<del>Linda</del>	<del>Siperstein</del>	<del>\$39,875.50</del>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>On leave</i>

## Font.Underline

Description

A number that specifies whether the text should be underlined.

Value

Values are:

**0**—Not underlined

**1**—Underlined

Example

The following statement, when applied to the Font.Underline property of columns of employee information, causes the information to be underlined if the employee does not have health insurance:

```
If (bene_health_ins = 'N', 1, 0)
```

FOR INFO For pictures of this example, see "Font.Italic" on page 353.

## Font.Weight

Description

The weight of the text.

Value

Values are:

**100**—Thin

**200**—Extra light

**300**—Light

**400**—Normal

**500**—Medium

**600**—Semibold  
**700**—Bold  
**800**—Extrabold  
**900**—Heavy

---

**Most commonly used values**

The most commonly used values are 400 (Normal) and 700 (Bold). Your printer driver may not support all of the settings.

---

**Example**

The following statement, when applied to the Font.Weight property of columns of employee information, causes the information to be displayed in bold if the employee does not have health insurance:

```
If(bene_health_ins = 'N', 700, 400)
```

FOR INFO For pictures of this example, see "Font.Italic" on page 353.

## Format

**Description**

The display format for a column.

**Values**

A string specifying the display format.

**Example**

The following statement, when applied to the Format property of the Salary column, causes the column to display the word *Overpaid* for any salary greater than \$60,000 and *Underpaid* for any salary under \$60,000:

```
If(salary>60000, 'Overpaid', 'Underpaid')
```

---

**Edit Mask edit style change**

The Edit Mask edit style assigned to the salary column had to be changed. Because edit styles take precedence over display formats, it was necessary to change the edit style assigned to the salary column (an Edit Mask edit style) to the Edit edit style.

---



The following report shows the values in the salary column to be either the word *Underpaid* or *Overpaid*:

Employee Report 01/11/98							
Department ID	Employee ID	Employee First Name	Employee Last Name	Salary	Health Ins.	Life Ins.	Day Care
100							
	102	Fran	Whitney	Underpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	105	Matthew	Cobb	Overpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	160	Robert	Breault	Underpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	243	Natasha	Shishov	Overpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	247	Kurt	Driscoll	Underpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	249	Rodrigo	Guevara	Underpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	266	Ram	Gowda	Underpaid	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	278	Terry	Melkisetian	Underpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	316	Lynn	Pastor	Overpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	445	Kim	Lull	Overpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	453	Andrew	Rabkin	Overpaid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

## Height

Description

The height of the column or other object.

Value

An integer in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.

Example

The following statement causes the height of a rectangle to be .350 inches if the state column for the row has the value NY. Otherwise, the rectangle is .200 inches high:

```
If(state = 'NY',350, 200)
```

FOR INFO For more details and pictures, see "Example 4: changing the size and location of objects" on page 340.

## Pen.Color

Description

The color of the line or the outline of a graphic object.

Value

A number that specifies the color of the line or outline.

FOR INFO For information on specifying colors, see "Specifying colors" on page 364.

Example See the example for the Pen.Style property, next.

## Pen.Style

Description The style of the line or the outline of a graphic object.

Value Values are:

- 0—Solid
- 1—Dash
- 2—Dotted
- 3—Dash-dot pattern
- 4—Dash-dot-dot pattern
- 5—Null (no visible line)

Example In this example, statements check the employee's start date to see if the month is the current month or the month following the current month. Properties of a rectangle object placed behind the row of data are changed to highlight employees with months of hire that match the current month or the month following the current month.

The workspace includes columns of data and a rectangle behind the data. The rectangle has been changed to black in the following picture to make it stand out:

today()		Performance Review Reminder		
Department ID	Employee ID	Employee First Name	Employee Last Name	Start Date
dept_id	emp_id	emp_fname	emp_lname	start_date

The following statement is for the Pen.Color property of the line around the edge of the rectangle. If the month of the start date matches the current month or the next one, Pen.Color is set to light gray (12632256). If not, it's set to white (16777215), which means it will not show:

```
If(month( start_date ) = month(today())
or month( start_date ) = month(today())+1
or (month(today()) = 12 and month(start_date)=1),
12632256, 16777215)
```

The following statement is for the Pen.Style property of the rectangle. If the month of the start date matches the current month or the next one, Pen.Style is set to Solid (0). If not, it's set to NULL (5), which means it will not show:

```
If(month( start_date ) = month(today())
or month( start_date ) = month(today())+1
or (month(today()) = 12 and month(start_date)=1),
0, 5)
```

The Expressions property page also includes expressions for Brush.Color and Brush.Hatch.

FOR INFO For more about these properties, see "Brush.Hatch" on page 346.

The following is a portion of the resulting report. A rectangle with light gray cross-hatching highlights employees whose reviews are due soon. The line enclosing the rectangle is Light Gray and uses the pen style Solid (0):

01/11/98		Performance Review Reminder			
Department ID	Employee ID	Employee First Name	Employee Last Name	Start Date	
400	888	Doug	Charlton	03/10/1991	
200	902	Moira	Kelly	04/01/1991	
200	913	Ken	Martel	04/16/1991	
500	921	Charles	Crowley	04/22/1991	
200	930	Ann	Taylor	05/08/1991	
200	949	Pamela	Savarino	05/08/1991	
100	958	Thomas	Sisson	07/16/1991	
400	992	Joyce	Butterfield	08/13/1991	
500	1013	Joseph	Barker	09/10/1991	
200	1021	Paul	Sterling	10/28/1991	
200	1039	Shih Lin	Chao	11/11/1991	
400	1062	Barbara	Blaikie	11/20/1991	
100	1090	Susan	Srnith	12/13/1991	
200	1101	Mark	Preston	01/09/1992	Review due this month
200	1142	Alison	Clark	01/19/1992	Review due this month
100	1157	Hing	Soo	01/29/1992	Review due this month
200	1162	Kevin	Goggin	02/03/1992	Review due next month
400	1191	Matthew	Bucceri	02/12/1992	Review due next month

## Pen.Width

Description

The width of the line or the outline of a graphic object.

Value

An integer in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.

**Example** The following statement causes the width of a line to be .05 inches if the state column for the row has the value NY. Otherwise, the line is .01 inches wide:

```
If(state = 'NY', 50, 10)
```

**FOR INFO** For more details and pictures, see "Example 4: changing the size and location of objects" on page 340.

## Pointer

**Description** The image used for the mouse pointer when the pointer is over the specified object.

**Value** A string that specifies a value of the Pointer enumerated data type or the name of a cursor file (CUR) used for the pointer.

Values of the Pointer enumerated data type are:

```
Arrow!  
Cross!  
HourGlass!  
IBeam!  
Icon!  
Size!  
SizeNESW!  
SizeNS!  
SizeNWSE!  
SizeWE!  
UpArrow!
```

**Example** The following condition, entered for the Pointer property of every object in a row of expense data, changes the pointer to the PowerBuilder column every time the value in the expense column exceeds \$100,000. Note that the pointer has no meaning in a printed report. The pointer is for use on the screen display of a report:

```
If(expense > 100000, 'pbcolumn.cur', 'arrow!')
```

## Visible

**Description** Whether the object is visible in the report.

**Value** Values are:

0—Not visible

1—Visible

### Example

The following statement is for the Visible property of a text object with the words On Leave located to the right of columns of employee information. The statement says that if the current employee's status is L, which means On Leave, the text object is visible. Otherwise it's invisible:

```
If(status = 'L', 1, 0)
```

### The status column must be retrieved

The status column must be included in the data source even though it does not appear in the report itself.

The workspace includes the text object at the right-hand end of the detail line. The text object is visible at execution time only if the value of the status column for the row is L.

The following is a portion of the resulting report. It shows two employees who are On Leave. The text object is visible only for the two employees on leave:

102	Fran	Whitney	\$45,700.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
105	Matthew	Cobb	\$62,000.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
160	Robert	Breault	\$57,490.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
243	Natasha	Shishov	\$72,995.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
247	Kurt	Driscoll	<del>\$48,023.60</del>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<i>On leave</i>
249	Rodrigo	Guevara	\$42,998.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
266	Ram	Gowda	\$59,840.00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
278	Terry	Melkisetian	\$48,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
316	Lynn	Pastor	\$74,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
445	Kim	Lull	\$87,900.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
453	Andrew	Rabkin	\$64,500.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
479	Linda	Siperstein	<del>\$39,876.50</del>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>On leave</i>

For the employees on leave, four columns of information show as crossed out. This is specified with property conditional expressions for those columns (described in the section on "Font.Strikethrough" on page 354).

## Width

Description

The width of the object.

Value	An integer in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.
Example	The following statement causes the width of a rectangle to be .900 inches if the state column for the row has the value NY. Otherwise, the rectangle is 2.000 inches wide:  <pre>If(state = 'NY', 900, 2000)</pre> FOR INFO For more details and pictures, see "Example 4: changing the size and location of objects" on page 340.

## X

Description	The distance of the object from the left edge of the report. At execution time, the distance from the left edge of the report is calculated by adding the margin to the x value.
Value	An integer in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.
Example	The following statement causes a rectangle to be located 6.250 inches from the left if the state column for the row has the value NY. Otherwise, the rectangle is 4.000 inches from the left:  <pre>If(state = 'NY', 6250, 4000)</pre> FOR INFO For more details and pictures, see "Example 4: changing the size and location of objects" on page 340.

## X1, X2

Description	The distance of each end of the line from the left edge of the report as measured in the workspace. At execution time, the distance from the left edge of the report is calculated by adding the margin to the x1 and x2 values.
Value	Integers in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.

**Example** The following statements for the X1 and X2 properties of a line cause the line to extend from 6.250 to 7.150 inches from the left if the state column for the row has the value NY. Otherwise, the line extends from 4.000 to 6.000 inches from the left:

```
If(state = 'NY', 6250, 4000)
If(state = 'NY', 7150, 6000)
```

**FOR INFO** For more details and pictures, see "Example 4: changing the size and location of objects" on page 340.

## Y

**Description** The distance of the object from the top of the band in which the object is located.

**Value** An integer in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.

**Example** **FOR INFO** For information, see "Example 4: changing the size and location of objects" on page 340.

## Y1, Y2

**Description** The distance of each end of the specified line from the top of the band in which the line is located.

**Value** Integers in the unit of measure specified for the report. Units of measure include PowerBuilder units, thousandths of an inch (1000 = 1 inch), thousandths of a centimeter (1000 = 1 centimeter), or pixels.

**Example** The following statements for the Y1 and Y2 properties of a line cause the line to be located .400 inches (Y1 and Y2 equal .400 inches) from the top of the detail band, if the state column for the row has the value NY. Otherwise, the line is located .250 inches (Y1 and Y2 equal .250 inches) from the top of the detail band:

```
If(state = 'NY', 400, 250)
If(state = 'NY', 400, 250)
```

**FOR INFO** For more details and pictures, see "Example 4: changing the size and location of objects" on page 340.

## Specifying colors

You specify a color by specifying a number that represents the color. You can specify the number explicitly or by using an expression that includes the RGB (*r, g, b*) function.

For the numbers and expressions that specify common colors, see the table below.

How the number is calculated

The formula for combining color values into a number is:

$$red + 256 * green + 256 * 256 * blue$$

where the amount of each primary color (red, green, and blue) is specified as a value from 0 to 255.

The RGB (*r, g, b*) function calculates the number from the amounts of red, green, and blue specified.

Sample numeric calculation

To create cyan, you use blue and green, but no red. If you wanted to create the most saturated (bright) cyan, you would use maximum amounts of blue and green in the formula, which is indicated by the number 255 for each. The following statements show the calculation:

$$\begin{aligned} &red + 256 * green + 256 * 256 * blue \\ &0 + 256 * 255 + 256 * 256 * 255 \\ &0 + 65280 + 16711680 \\ &16776960 \end{aligned}$$

Sample expression using the RGB function

The following expression also specifies the brightest cyan:

$$\begin{aligned} &RGB(r, g, b) \\ &RGB(0, 255, 255) \\ &16776960 \text{ (The RGB function returns this number)} \end{aligned}$$

Notice that the expression specifies the maximum for green and blue (255) and 0 for red. To specify cyan, entering the expression **RGB(0, 255, 255)** is the same as entering the number **16776960**.

Numbers and expressions to enter for the common colors

This table shows the numbers and expressions to enter for some common colors. The number and expression for a color are equivalent. You can use either:

Color	Expression to enter	Number to enter	How the number is calculated
Black	RGB (0, 0, 0)	0	0 (no color)



<b>Color</b>	<b>Expression to enter</b>	<b>Number to enter</b>	<b>How the number is calculated</b>
Blue	RGB (0, 0, 255)	16711680	$256*256*255$ (blue only)
Cyan	RGB (0, 255, 255)	16776960	$256*255 + 256*256*255$ (green and blue)
Dark Green	RGB (0, 128, 0)	32768	$256*128$ (green only)
Green	RGB (0, 255, 0)	65280	$256*255$ (green only)
Light Gray	RGB (192, 192, 192)	12632256	$192 + 256*192 + 256*256*192$ (some red, green, and blue in equal amounts)
Lighter Gray	RGB (224, 224, 224)	14737632	$224 + 256*224 + 256*256*224$ (some red, green, and blue in equal amounts)
Lightest Gray	RGB (240, 240, 240)	15790320	$240 + 256*240 + 256*256*240$ (some red, green, and blue in equal amounts)
Magenta	RGB (255, 0, 255)	16711935	$255 + 256*256*255$ (red and blue)
Red	RGB (255, 0, 0)	255	255 (red only)
White	RGB (255, 255, 255)	16777215	$255 + 256*255 + 256*256*255$ (red, green, and blue in equal amounts at the maximum of 255)
Yellow	RGB (255, 255, 0)	65535	$255 + 256*255$ (red and green)



# Using Nested Reports

About this chapter

This chapter provides information about creating reports that have other reports nested in them.

Contents

<b>Topic</b>	<b>Page</b>
About nested reports	368
Creating a report using the Composite presentation style	372
Placing a nested report in another report	374
Working with nested reports	379

# About nested reports

A **nested report** is a report in another report.

There are two ways to create reports that have nested reports:

- ◆ Create a composite report using the Composite presentation style
- ◆ Place a nested report in another report

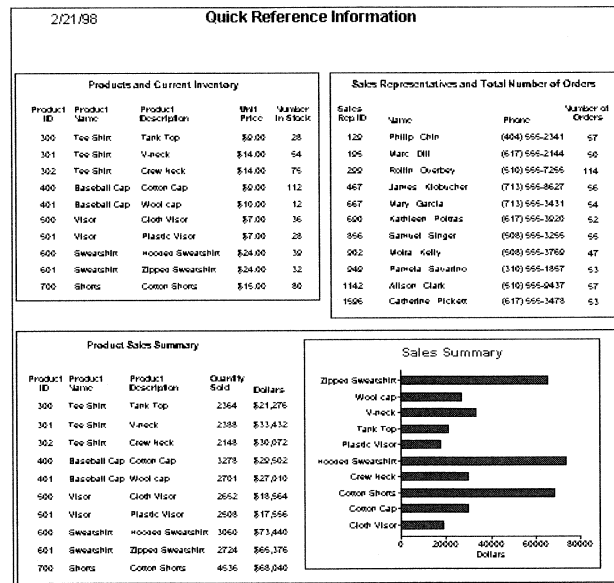
About creating a composite report

You can choose the Composite presentation style to create a new report that consists entirely of one or more nested reports. This type of report is called a **composite** report. A composite report is a container for other reports.

You can use composite reports to print more than one report on a page.

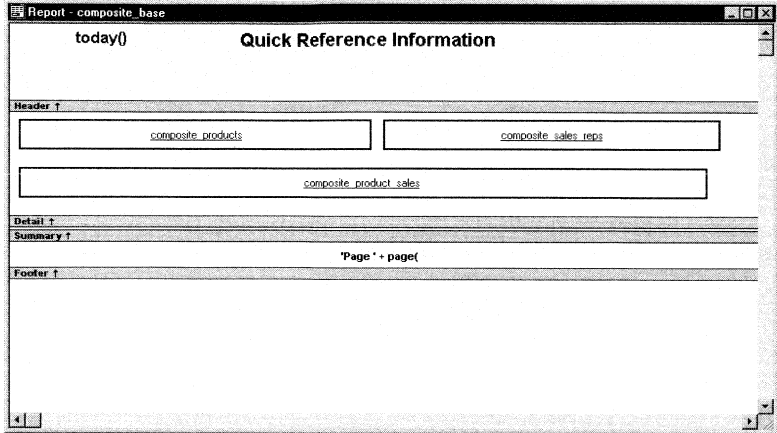
Composite report

For example, the following composite report consists of three tabular reports. One of the tabular reports includes a graph:



Composite report in the workspace

In the workspace, you see three boxes that represent the individual tabular reports that are included in the composite report. The only additional objects in this example are a title, date, and page number:



About placing a nested report within another report

You can place one or more reports within another report. The report you place is called the nested report. You can place a nested report in any type of report except crosstab. Most of the time you will place nested reports in freeform or tabular reports.

Often, the information in the nested report depends on information in the report in which it is placed (the **base report**). The nested report and the base report are related to each other by some common data. The base report and the nested report have a master/detail relationship.

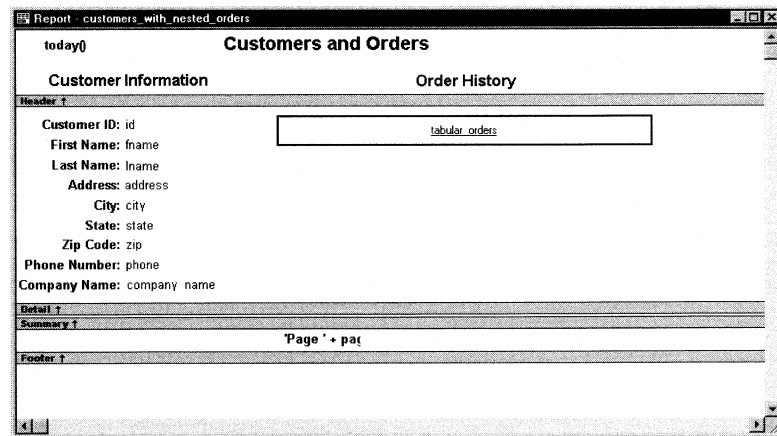
Freeform report with a related nested report

For example, the following freeform report lists all information about a customer and then includes a related nested report (it happens to be a tabular report). The related nested report lists every order that the customer has ever placed. The base report supplies the customer ID to the nested report, which requires a customer ID as a retrieval argument. This is an example of a master/detail relationship—one customer has many orders:

2/21/98		Customers and Orders					
Customer Information			Order History				
Customer ID:	105						
First Name:	Laura						
Last Name:	McCarthy						
Address:	1210 Highway 36						
City:	Carmel						
State:	IN						
Zip Code:	46032						
Phone Number:	(317) 555-8437						
Company Name:	Arno & Sons						
Sales Order ID	Order Date	Sales Rep ID	Line #	Product ID	Quantity	Date Shipped	
2006	09/28/95	299	1	300	48	09/28/95	
2344	03/30/95	195	1	501	36	03/31/95	
2454	06/16/95	299	1	501	36	06/17/95	
2568	09/21/95	856	1	600	36	09/22/95	
			2	601	36	09/22/95	

What you see in the workspace

In the workspace, you see everything in the base report plus a box that represents the related nested report:



What's the difference?

There are two important differences between nesting using the Composite style and nesting a report within a base report.

**Data sources** The composite report does not have a data source—it is just a container for nested reports. In contrast, a base report with a nested report in it has a data source. The nested report has its own data source.

**Related nesting** The composite report cannot be used to relate reports to each other in the database sense. One report cannot feed a value to another report, which is what happens in a master/detail report. If you want to relate reports to each other so that you can create a master/detail report, you need to place a nested report within a base report.

#### How retrieval works

When you preview (run) a composite report, InfoMaker retrieves all the rows for one nested report, and then for another nested report, and so on until all retrieval is complete.

When you preview (run) a report with another related report nested in it, InfoMaker retrieves all the rows in the base report first. Then InfoMaker retrieves the data for all nested reports related to the first row. Next, InfoMaker retrieves data for nested reports related to the second row, and so on, until all retrieval is complete for all rows in the base report.

**FOR INFO** For information about efficiency and retrieval, see "Supplying retrieval arguments to relate a nested report to its base report" on page 383.

#### Limitations on nesting reports

For the most part you can nest the various types of report styles. However, limitations apply to two of them.

**Crosstabs** You cannot place a crosstab with retrieval arguments within another report as a related nested report. However, you can include a crosstab in a Composite report.

**RichText reports** You cannot nest a RichText report in any way. You cannot place a RichText report in another report and you cannot include a RichText report in a Composite report.

## Creating a report using the Composite presentation style

❖ **To create a report using the Composite presentation style:**

- 1 Click the Report button in the PowerBar.

The Select Report dialog box displays listing reports in the selected library.

- 2 Click the New button.

The New Report dialog box displays.

- 3 Click the Composite presentation style.

Notice that all the data sources become unavailable (grayed out) when you click Composite. Composite reports do not have data sources. They include other reports, which have their own data sources.

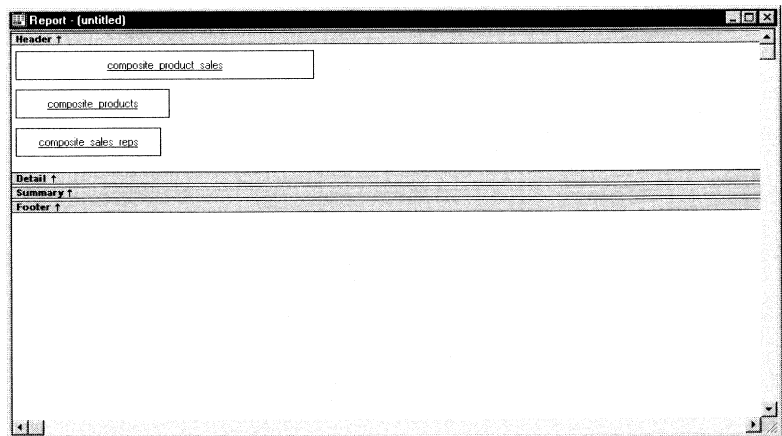
- 4 Click OK.

The Select Reports dialog box displays listing all reports that are in the current library.

- 5 Click the reports you want to include in the composite report and then click OK.

- 6 InfoMaker places boxes for the selected reports in the workspace.

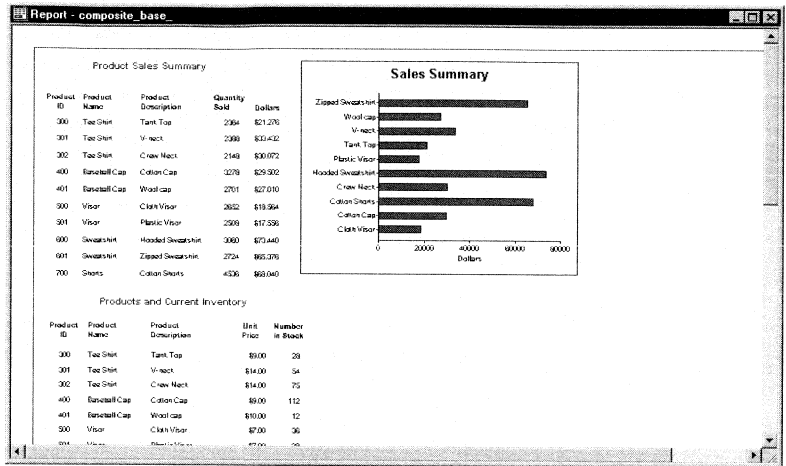
In this example, you see three reports:



- 7 Select File>Save from the menu bar and assign a name to the report.



- 8 Click the Preview button to see what your report looks like:



### Previewing composite reports differs from previewing other types of reports

The Rows>Filter, Rows>Import, and Rows>Sort menu items are grayed when you preview a composite report. You cannot choose these options. If you want to use any of these options, you can go back to the workspace and access the nested report(s), where these options are available in preview.

- 9 Click the Preview button to return to the workspace and continue to enhance the composite report.

## Placing a nested report in another report

When you place a nested report in another report, the two reports can be independent of each other or they can be related in the database sense by sharing some common data such as a customer number or a department number. If the reports are related, you need to do some extra things to both the base report and the related nested report.

Usually when you place a report within a report rather than create a composite report, you want to relate the reports. Those instructions are first.

## Placing a related nested report in another report

Typically, a related nested report provides the details for a master report. For example, a master report might provide information about customers. A related nested report placed in the master report could provide information about all the orders that belong to each customer.

### ❖ To place a related nested report in another report:

- 1 Create the nested report that you plan to place in the base report.
- 2 Define a retrieval argument for the nested report.

For example, suppose the nested report lists orders and you want to list orders for a particular customer. To define a retrieval argument you would:

- ◆ Select Design>Edit Data Source to go to the Select painter.
- ◆ Select Design>Retrieval Arguments from the menu bar in the Select painter.
- ◆ Define a retrieval argument in the Specify Retrieval Arguments dialog box. In the example, *customerID* is the name assigned to the retrieval argument.

- Specify the retrieval argument in a WHERE clause for the SELECT statement.

The WHERE clause in this example tells the DBMS to retrieve rows where the value in the column *cust\_id* equals the value of the argument *:customerid*:

Sort	Where	Group	Having	Compute	Syntax
	Column		Operator		Value
	"sales_order"."cust_id"		=		:customerid
					Logical

At this point, when you preview (run) the report, you are prompted to enter a value for *:customerid*. Later in these steps you will specify that the base report supply the values for *:customerid* instead of prompting for values.

- Open or create the report you want to be the base report.

In the example, the base report is one that lists customers and has a place for the order history of each customer:

The screenshot shows a report window titled "Report - customers\_with\_nested\_orders". The report content is as follows:

```

today()
Customers and Orders
Customer Information                                Order History
Header
Customer ID: id
First Name: fname
Last Name: lname
Address: address
    City: city
    State: state
    Zip Code: zip
Phone Number: phone
Company Name: company name
Detail
Summary
Page 1 of page
Footer
  
```

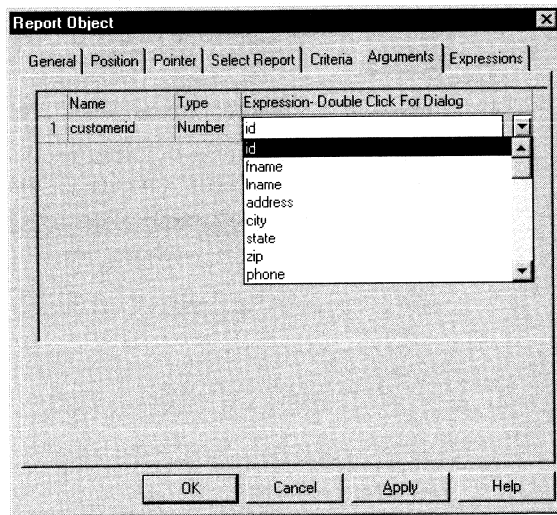
- Click the Nested Report button in the Objects dropdown toolbar.  
*or*  
Select Objects>Report from the menu bar.
- Click in the workspace where you want to place the report.  
The Report Object property sheet displays with the Select Report property page on top. The Select Report property page lists defined reports in the current library.
- Select the report you want and click Apply.

- 8 Select the Arguments tab of the property sheet.

The Arguments property page lists arguments defined for the nested report and provides a place for you to specify how information from the base report will be used to supply the values of arguments to the nested report.

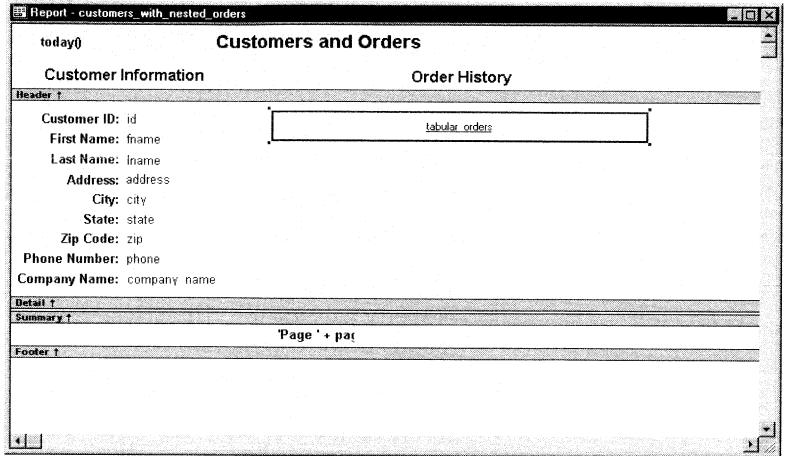
- 9 Select the base report column that will supply the argument's value from the drop down listbox and click OK.

You can also double-click to enter an expression that will be used to supply a value for the argument. In the example, the column named *id* from the base report will supply the value for the argument *:customerid* in the nested report:

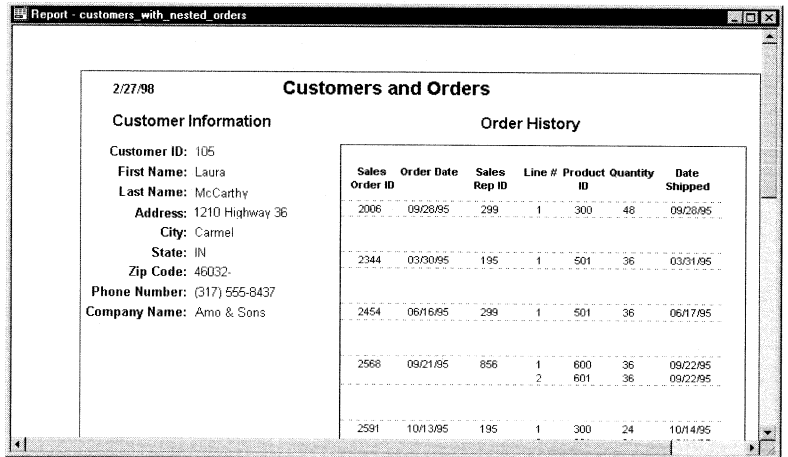


- 10 Click OK.

A box representing the report displays in the workspace, as shown in this example:



- 11 Select File>Save from the menu bar and assign a name to the report.
- 12 Click the Preview button to see what your report looks like:



FOR INFO For more information about what you can do while previewing a report, see Chapter 5, "Enhancing Reports".

- 13 Click the Preview button to return to the workspace and continue to enhance the report.

## Placing an unrelated nested report in another report

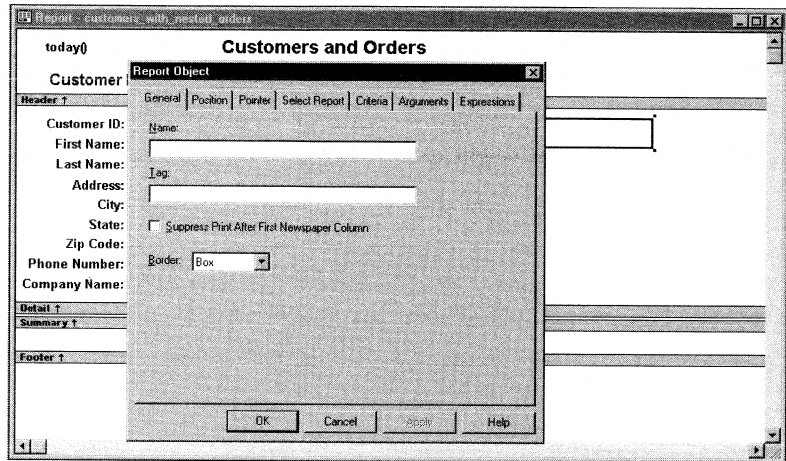
When you place an unrelated nested report in a base report, the entire nested report will appear with *each* row of the base report.

❖ **To place an unrelated nested report in another report:**

- 1 Open the report you want to be the base report.
- 2 Click the Nested Report button in the Objects dropdown toolbar.  
*or*  
Select Objects>Report from the menu bar.
- 3 Click in the workspace where you want to place the report.  
The Report Object property sheet displays with the Select Report property page on top. The Select Report property page lists defined reports in the current library.
- 4 Select the report you want and click OK.  
A box representing the report displays in the workspace.
- 5 Select File>Save from the menu bar and assign a name to the report.
- 6 Click the Preview button to see what your report looks like.  
FOR INFO For more information about what you can do while previewing a report, see Chapter 5, "Enhancing Reports".
- 7 Click the Preview button to return to the workspace and continue enhancing the report.

## Working with nested reports

When you use nested reports either in composite reports or in other base reports, several enhancements and options are available. An easy way to see what you can do is to display the property sheet for the nested report. The following screen shows the property sheet for a related nested report:



Many of the options in the property sheet are described in Chapter 5, "Enhancing Reports"—not here. For example, using borders on nested reports is like using borders on any object.

This section describes activities that apply only to nested reports or that have special meaning for nested reports. It covers:

- ◆ "Adjusting nested report width" next
- ◆ "Using the Autosize Height option for nested reports" on page 380
- ◆ "Changing a nested report from one report to another" on page 381
- ◆ "Modifying the contents of a nested report" on page 382
- ◆ "Adding another nested report to a composite report" on page 382
- ◆ "Supplying retrieval arguments to relate a nested report to its base report" on page 383
- ◆ "Specifying criteria to relate a nested report to its base report" on page 385
- ◆ "Using the Slide option for a nested report" on page 387
- ◆ "Using the Start On New Page option (composite only)" on page 387
- ◆ "Using the Trail the Footer option (composite only)" on page 387

## Adjusting nested report width

When you preview a report with nested reports, you may find that the width of the nested report is unacceptable. This can happen, for example, if you change the design of the nested report or if you use newspaper columns in a nested report.

### ❖ To adjust report width:

- 1 In the workspace, position the pointer near a vertical edge of the nested report and press the left mouse button.
- 2 Drag the edge to widen the nested report.
- 3 Click the Preview button to check the new width.

## Using the Autosize Height option for nested reports

Autosize Height must be on for all nested reports except graphs. This option ensures that the height of the nested report can change to accommodate the rows that are returned.



This option is on by default for all nested reports except graphs. Usually there is no reason to change it. If you do want to force a nested report to have a fixed height, you can turn this option off.

Note that the detail band also has an Autosize Height option. The option is on by default and must be on for the Autosize Height option for the nested report to work properly.

❖ **To change the Autosize Height option in a nested report:**

- 1 Select Properties from the nested report's popup menu and then select the Position tab.
- 2 Select/clear the Autosize Height checkbox.

---

**If you put a nested report in a band other than the detail band**

Bands other than the detail band do not have an Autosize Height option. If you place a nested report in a band other than the detail band, you must size the band itself to hold the report.

---

## Changing a nested report from one report to another

When you are working with nested reports, you may want to change the nested report that is used. For example, you may work on several versions of a nested report and need to update the version of the nested report that the composite or base report uses.

❖ **To change the nested report:**

- 1 Select Properties from the nested report's popup menu and then select the Select Report tab.

The Select Report property page lists all the reports in the current library.

- 2 Select the report you want to use and click OK.

The name of the report that displays in the box in the workspace changes to the new one.

## Modifying the contents of a nested report

While you are working with nested reports, you may want to modify the contents of the nested report. You can do this directly from the composite report or base report that contains the nested report.

❖ **To modify the contents of a nested report from the composite report or base report:**

1 Position the pointer on the nested report whose contents you want to modify and display the popup menu.

2 Select Modify Report from the popup menu.

The nested report opens and displays in the Report painter workspace. Both the composite or base report and the nested report are open.

3 Modify the report.

4 Select File>Close from the menu bar.

You are prompted to save your changes.

5 Click OK.

You return to the composite report or to the base report that includes the nested report.

## Adding another nested report to a composite report

After you have created a composite report, you may want to add another report.

**FOR INFO** For information on adding a nested report to a report (not a composite report), see "Placing a related nested report in another report" on page 374 or "Placing an unrelated nested report in another report" on page 378.

❖ **To add another nested report to a composite report:**

1 Open the composite report.

2 Click the Report button in the Objects dropdown toolbar.

*or*

Select Objects>Report from the menu bar.

3 Click in the workspace where you want to place the report.

The Report Object property sheet displays with the Select Report property page on top. The Select Report property page lists defined reports in the current library.

- 4 Select the report you want and click OK.  
A box representing the report displays in the workspace.

## Supplying retrieval arguments to relate a nested report to its base report

The most efficient way to relate a nested report to its base report is to use retrieval arguments. If your nested report has arguments defined, you use the process described in this section to supply the retrieval argument value from the base report to the nested report. (The process described is part of the whole process covered in "Placing a related nested report in another report" on page 374.)

Why retrieval arguments are efficient

Some DBMSs have the ability to bind input variables in the WHERE clause of the SELECT statement. When you use retrieval arguments, a DBMS *with this capability* sets up placeholders in the WHERE clause and compiles the SELECT statement *once*. InfoMaker retains this compiled form of the SELECT statement for use in subsequent retrieval requests.

Requirements for reusing the compiled SELECT statement

To enable InfoMaker to retain and reuse the compiled SELECT statement:

- ◆ The database interface must support SQL caching.
- ◆ You must enable caching in the database profile. You should set the SQLCache DBParm parameter to the number of levels of nesting plus 5.

**FOR INFO** For more information, see the description of the SQLCache DBParm parameter in *Connecting to Your Database*.

---

### **Nested reports in composite reports**

If the base report is a composite report, you need to define retrieval arguments for the composite report before you can supply them to the nested report.

Display the properties sheet for the composite report and select the Retrieval Arguments tab. Then define the retrieval arguments that the nested report needs, taking care to specify the correct type.

---

#### **❖ To supply a retrieval argument value from the base report to the nested report:**

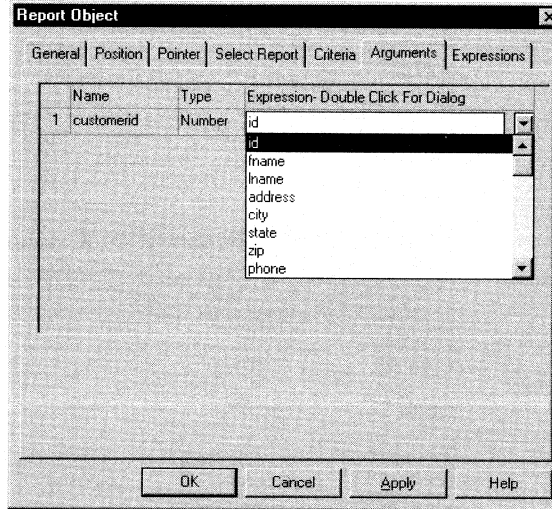
- 1 Make sure that the nested report has been set up to take one or more retrieval arguments.  
  
FOR INFO See "Placing a nested report in another report" on page 374.
- 2 Select Properties from the nested report's popup menu and then select the Arguments tab.

The Arguments property page lists arguments defined for the nested report and provides a place for you to specify how information from the base report will supply the value of the argument to the nested report.

- 3 Select the base report column that will supply the argument's value from the drop down listbox and click OK.

You can also click the dropdown arrow to display a list of the columns in the base report. You can double-click to enter an expression that will supply a value for the argument.

In the example, the column named *id* from the base report will supply the value for the argument *:customerid* in the nested report:



You return to the workspace. When you run the report now, you will not be prompted for retrieval argument values for the nested report. The base report will supply the retrieval argument values automatically.

## Specifying criteria to relate a nested report to its base report

If you do not have arguments defined for the nested report and if database efficiency is not an issue, you can place a nested report in another report and specify criteria to pass values to the related nested report.

How the DBMS processes SQL if you use the specify criteria technique

If you use the specify criteria technique, the DBMS repeatedly recompiles the SELECT statement and then executes it. The recompilation is necessary for each possible variation of the WHERE clause.

❖ **To specify criteria to relate a nested report to its base report:**

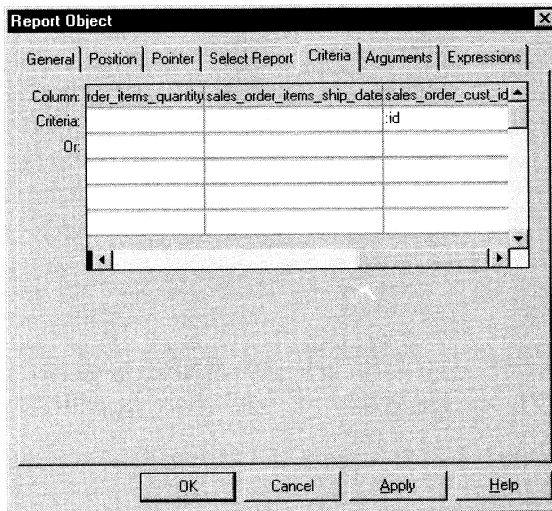
- 1 Select Properties from the nested report's popup menu and then select the Criteria tab.

The Criteria property page lists all the columns retrieved for the nested report and provides a place for you to specify how information from the base report will supply the retrieval criteria to the nested report.

- 2 Enter the retrieval criteria and click OK.

The rules for specifying criteria are the same as for specifying criteria in the Quick Select data source. Multiple criteria in one criteria line are ANDed together. Criteria entered on separate lines are ORed together.

In this example, the customer ID (the *id* column) is the retrieval criteria being supplied to the nested report. Notice that the *id* column is preceded by a colon (:), which is required:



You return to the workspace. When you run the report now, InfoMaker retrieves rows in the nested report based on the criteria you have specified. In the example, the customer ID column in the base report determines which rows from the *sales\_order* table are included for each customer.

## Using the Slide option for a nested report

InfoMaker determines the appropriate Slide options when positioning the nested report(s) and assigns default values. Usually you should not change the default values:

- ◆ The Slide Left option is on by default for grid and crosstab style reports and off by default for all others. Having Slide Left on for grid and crosstab ensures that these reports break horizontally on whole columns and not in the middle of a column.
- ◆ The Slide Up All Above/Up Directly Above options ensure that the nested report uses just as much vertical space as it needs. One of these options is on by default for all nested reports.

**FOR INFO** For more information, see "Sliding objects to remove blank space in a report" on page 240.

## Using the Start On New Page option (composite only)

The Start On New Page option forces a new page for a nested report used *in a composite report*. By default, this option is off.

❖ **To specify that a nested report in a composite report should begin on a new page:**

- 1 Select Properties from the nested report's popup menu and then select the General tab.
- 2 Select the Start On New Page checkbox.

A checkmark displays indicating the option is selected.

## Using the Trail the Footer option (composite only)

The Trail the Footer option controls the placement of the footer for the last page of a nested report *in a composite report*. By default, this option is on. The footer appears directly under the contents of the nested report and not at the bottom of the page.

❖ **To specify that the footer should appear at the bottom of the page:**

- 1 Select Properties from the nested report's popup menu and then select the General tab.

2 Clear the Trail the Footer checkbox.

The checkmark next to the option disappears, indicating the option is no longer selected. The footer will appear at the bottom of the page on all pages of the nested report, including the last page. Note that if another nested report begins on the same page, the footer from the earlier report might be misleading or confusing.



About this chapter

This chapter describes how to build and use graphs in InfoMaker.

Contents

<b>Topic</b>	<b>Page</b>
Overview of graphs	390
Using graphs in reports	396
Using the Graph presentation style	414
Defining a graph's properties	415

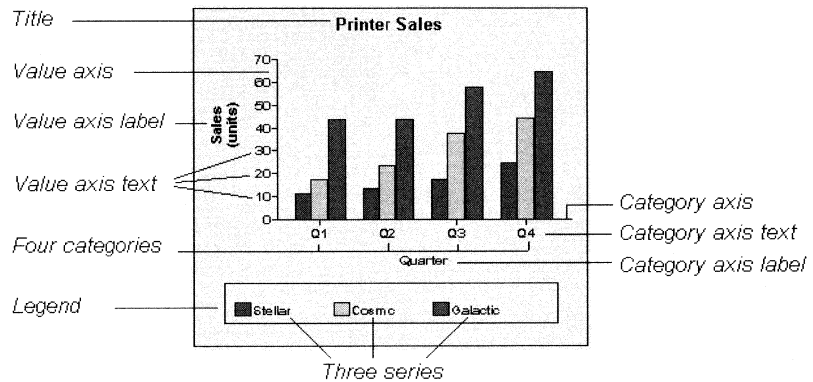
## Overview of graphs

Often the best way to display information is graphically. You can add a graph to a report and work with it there, or you can use the Graph presentation style to create a standalone graph. You can use graphs to supplement the numbers in a report or you can replace numbers with a graph.

InfoMaker provides many types of graphs and allows you to customize your graphs in many ways. Probably most of your use of graphs will be in a report—the source of the data for your graphs will be the database.

## Parts of a graph

Here is a column graph created in InfoMaker that contains most major parts of a graph. It shows quarterly sales of three products: Stellar, Cosmic, and Galactic printers:



## How data is represented

Graphs display data points. To define graphs, you need to know how the data is represented. InfoMaker organizes data into three components:

Component	Meaning
Series	<b>A set of data points</b> Each set of related data points makes up one series. In the preceding graph, there is a series for Stellar sales, another series for Cosmic sales, and another series for Galactic sales. Each series in a graph is distinguished by color, pattern, or symbol

<b>Component</b>	<b>Meaning</b>
Categories	<b>The major divisions of the data</b> Series data are divided into categories, which are often non-numeric. In the preceding graph, the series are divided into four categories: Q1, Q2, Q3, and Q4. Categories represent values of the independent variable(s)
Values	The values for the data points (dependent variables)

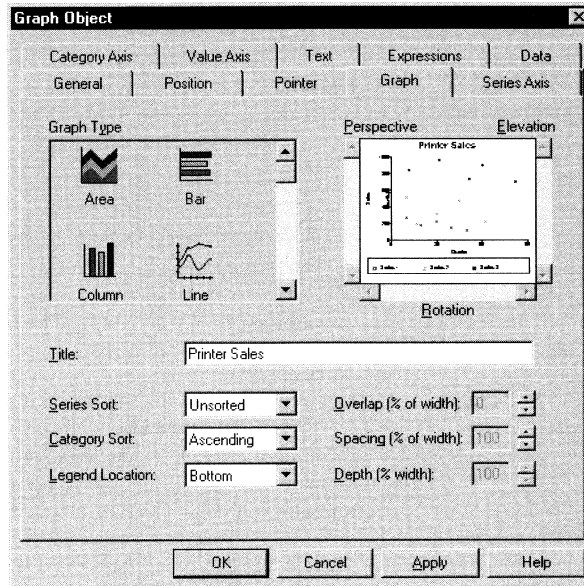
## Organization of a graph

The following table lists the parts of a typical graph:

<b>Part of graph</b>	<b>What it is</b>
Title	An optional title for the graph. The title appears at the top of the graph
Value axis	The axis of the graph along which the values of the dependent variable(s) are plotted. In a column graph, as shown in the preceding graph, the Value axis corresponds to the y axis in an XY presentation. But in other types of graphs, such as a bar graph, the Value axis can be along the x dimension
Category axis	The axis along which are plotted the major divisions of the data, representing the independent variable(s). In the preceding graph, the Category axis corresponds to the x axis. It plots four categories: Q1, Q2, Q3, and Q4. These form the major divisions of data in the graph
Series	A set of data points. There are three series in the preceding graph: Stellar, Cosmic, and Galactic. In bar and column charts, each series is represented by bars or columns of one color or pattern
Series axis	The axis along which the series are plotted in three-dimensional (3D) graphs
Legend	An optional listing of the series. The preceding graph contains a legend that shows how each series is represented in the graph

## Types of graphs

InfoMaker provides many graph types for you to choose from. You choose the type from the Graph property page of the Graph Object property sheet:



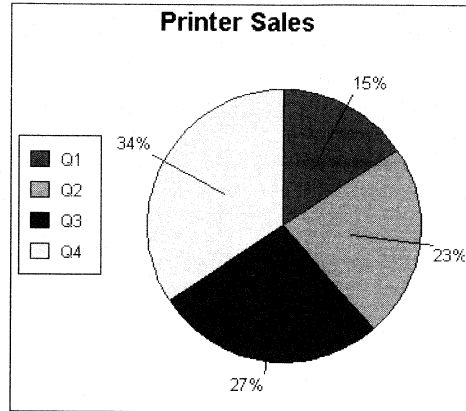
### Area, bar, column, and line graphs

Area, bar, column, and line graphs are conceptually very similar. They differ only in how they physically represent the data values—whether they use areas, bars, columns, or lines to represent the values. All other properties are the same. Typically you use area and line graphs to display continuous data and use bar and column graphs to display noncontinuous data.

The only difference between a bar graph and a column graph is the orientation: in column graphs, values are plotted along the y axis and categories are plotted along the x axis. In bar graphs, values are plotted along the x axis and categories are plotted along the y axis.

## Pie graphs

Pie graphs typically show one series of data points with each data point shown as a percentage of a whole. The following pie graph shows the sales for Stellar printers for each quarter. You can easily see the relative values in each quarter. (InfoMaker automatically calculates the percentages of each slice of the pie.)



You can have pie graphs with more than one series if you want; the series are shown in concentric circles. Multiseries pie graphs can be useful in comparing series of data.

## Scatter graphs

Scatter graphs show *xy* data points. Typically you use scatter graphs to show the relationship between two sets of numeric values.

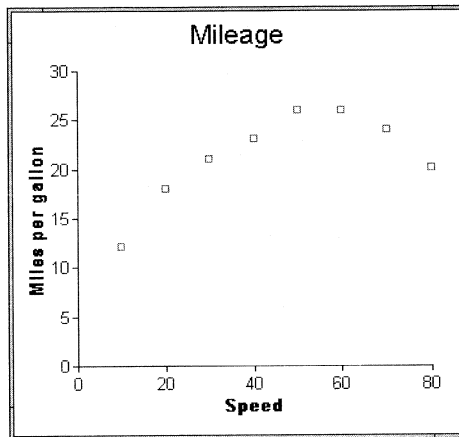
Scatter graphs do not use categories. Instead, numeric values are plotted along both axes—as opposed to other graphs, which have values along one axis and categories along the other axis.

For example, the following data shows the effect of speed on the mileage of a sedan:

Speed	Mileage
10	12
20	18
30	21
40	23

Speed	Mileage
50	26
60	26
70	24
80	20

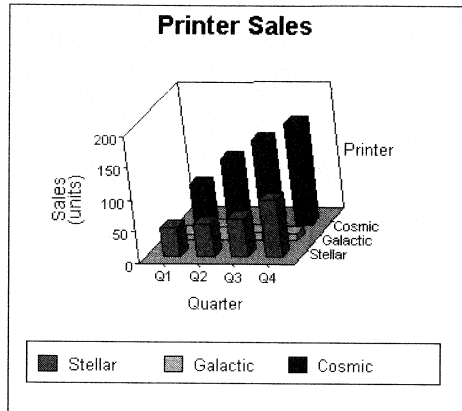
Here is the data in a scatter graph:



You can have multiple series of data in a scatter graph. In the above example, you might want to plot mileage versus speed for several makes of cars in the same graph.

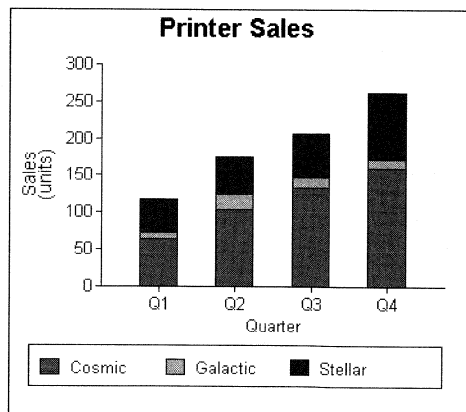
## Three-dimensional graphs

You can also create 3-dimensional (3D) graphs of area, bar, column, line, and pie graphs. In 3D graphs (except for 3D pie graphs), series are plotted along a third axis (the Series axis) instead of along the Category axis. You can specify the perspective to use to show the third dimension:



## Stacked graphs

In bar and column graphs, you can choose to stack the bars and columns. In stacked graphs, each category is represented as one bar or column instead of as separate bars or columns for each series:



## Using graphs in reports

Graphs in reports are dynamic

Graphs in reports are tied directly to the data that is in the report. As the data changes, the graph is automatically updated to reflect the new values.

Two techniques

You can use graphs in reports in two ways:

- ◆ By including a graph as an object in a report

Here you use a graph to enhance the display of information in a report, such as a tabular or freeform report. This technique is described next.

- ◆ By using the Graph presentation style

Here the entire report is a graph. The underlying data isn't visible. This technique is described in "Using the Graph presentation style" on page 414.

## Placing a graph in a report

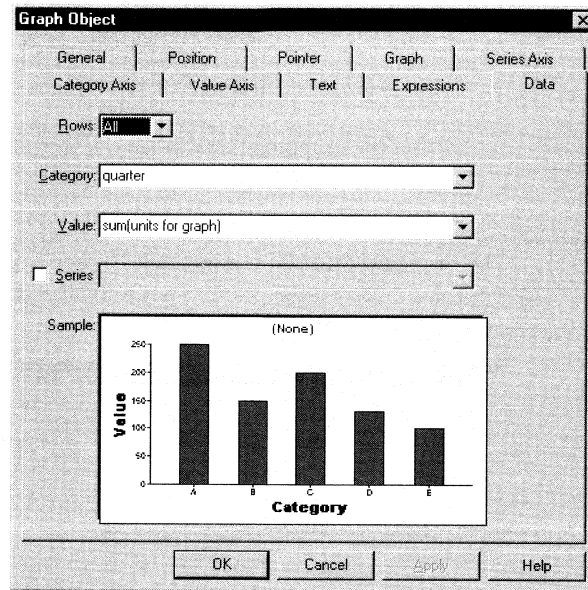
### ❖ To place a graph in a report:

- 1 Open the Report painter and select the report that will contain the graph. The Report painter workspace displays the report.
- 2 Click the Graph button in the Objects dropdown toolbar.



- 3 Click where you want the graph.

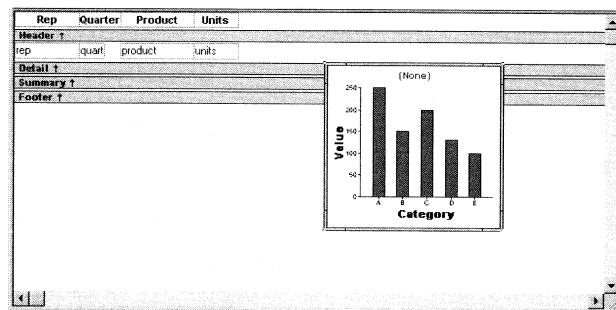
InfoMaker displays the Graph Object property sheet with the Data property page on top:



- 4 Specify which columns contain the data and click OK.

FOR INFO For more information, see "Associating data with a graph" on page 401.

You return to the painter workspace with a representation of the graph in place:



- 5 Specify the graph's properties using the graph's property sheet.

## Using the graph's property sheet

A graph has a property sheet in which you specify the data as well as the other properties of the graph.

❖ **To display the Graph Object property sheet:**

- ◆ Select Properties from the graph's popup menu.

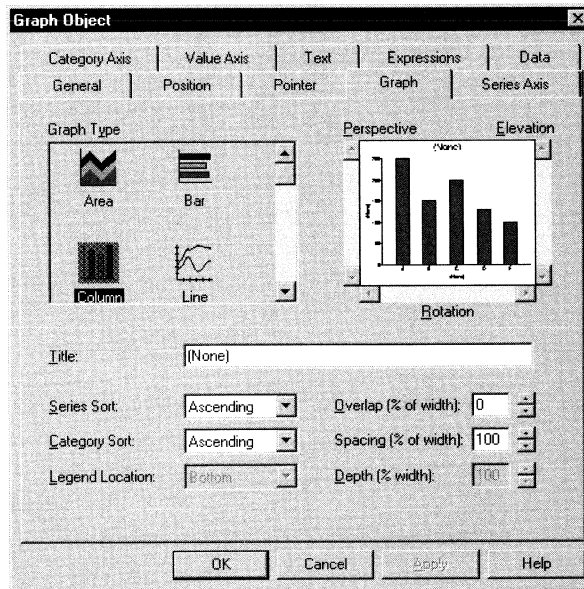
*or*

Select the graph and then select Design>Properties from the menu bar.

*or*

Select the graph and then click the Properties button.

The property sheet for a graph has 10 property pages in which you specify information about the graph. The following property sheet shows the Graph property page on top:



The following table lists the ten property pages on the Graph Object property sheet and describes what each property page specifies:

Property page	What it specifies
Category Axis	Labels, scale, information about major and minor divisions for the category axis
Data	Where to get the graph's data

Property page	What it specifies
Expressions	An expression that is evaluated at execution time to determine the value of a property of the graph
General	Various general graph properties, including border, graph colors, whether to size the graph to the full screen display, suppression in newspaper columns
Graph	Graph type, title, category and series sorting, legend location For 3D graphs, perspective, rotation, and elevation For bar graphs, overlap, spacing and depth of bars
Pointer	The pointer to use when the mouse is positioned over the graph
Position	The x,y location of the upper left corner of the graph, its width and height, sliding options, the layer in which the graph is to be positioned Whether the graph can be resized and moved during preview
Series Axis	Labels, scale, information about major and minor divisions for the series axis
Text	Text properties for text objects that display on the graph, including title, axis text, axis label, and legend Text properties include font, font style, font size, alignment, rotation, color, display expression, display format
Value Axis	Labels, scale, information about major and minor divisions for the value axis

## Changing a graph's position and size

When you initially place a graph in a report, it is in the foreground—it sits above the bands in the report. Unless you change this setting, the graph displays in front of any retrieved data.

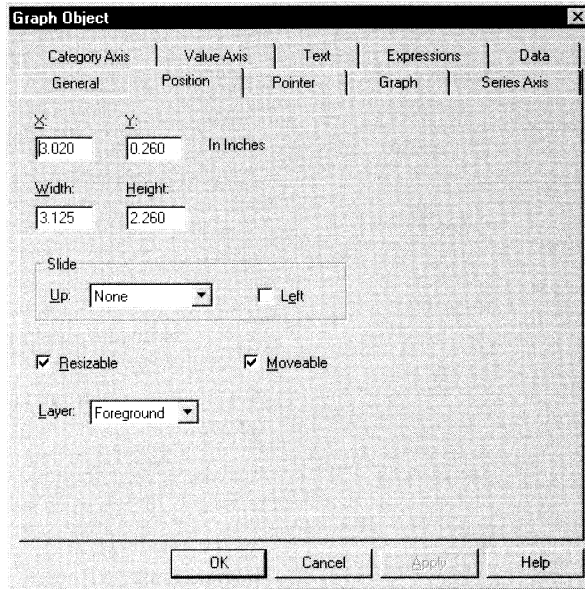
The initial graph is also movable and resizable, so you have complete flexibility as to the size and location of a graph when you run the report.

You can change these properties if you want.

❖ **To specify a graph's position and size:**

- 1 Select Properties from the graph's popup menu and then select the Position tab or the General tab.

The Position property page contains most of the options:



- 2 Select the settings for the following options on the Position property page:

Setting	Meaning
Layer	<p><i>Background</i> — The graph displays behind other elements in the report</p> <p><i>Band</i> — The graph displays in one particular band. If you choose this setting, you should resize the band to fit the graph. Often you will want to place a graph in the Footer band. As you scroll through rows in the report, the graph remains at the bottom of the screen as part of the footer</p> <p><i>Foreground</i> — (Default) The graph displays above all other elements in the report. Typically, if you choose this setting, you also make the graph movable so it won't obscure data while you display the report</p>
Moveable	The graph can be moved while displaying the report
Resizable	The graph can be resized while displaying the report

Setting	Meaning
Slide	The graph slides to the left or up to remove extra white space  FOR INFO For more information, see "Sliding objects to remove blank space in a report" on page 240.
X, Y	The location of the upper left corner of the graph
Width, Height	The width and height of the graph

- 3 Select the settings for the following options on the General property page:

Setting	Meaning
Size to Display	The graph fills the report and resizes when you resize the report. This setting is used with the Graph presentation style
Suppress After First	Don't repeat graph after the first column in a report using newspaper-style columns

## Associating data with a graph

When using a graph in a report, you associate axes of the graph with columns in the report.

In fact, the only way to get data into a graph in a report is through columns in the report. You cannot add, modify, or delete data in the graph except by adding, modifying, or deleting data in the report.

You can graph data from any columns retrieved into the report. The columns don't have to be displayed.

---

### About the examples

The process of specifying data for a graph is illustrated below using the Printer table in the Powersoft Demo Database.

---

#### ❖ To specify data for a graph:

- 1 If you have just placed a graph in the Report painter, the Graph Object property sheet is displayed. Otherwise, select Properties from the graph's popup menu. Select the Data tab.
- 2 Fill in the boxes as described below.

- 3 Click OK.

## Specifying which rows to include in a graph

The Rows dropdown listbox allows you to specify which rows of data are graphed at any one time:

Setting	Meaning
All	Graphs the data from all the rows that have been retrieved but not filtered or deleted (that is, the rows in the primary buffer of the report)
Page	Graphs only the data from the rows that are currently displayed on the page
Group n	Graphs only the data in the specified group (in a grouped report)

---

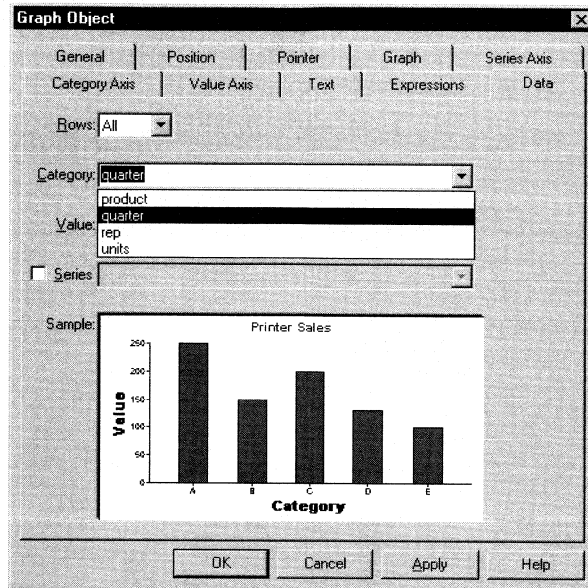
### If you select Group

If you are graphing data in the current group in a grouped report and may have several groups displayed at the same time, you should localize the graph in a group-related band in the workspace to ensure that it is clear which group the graph represents. Usually, the group header band is the most appropriate band.

---

## Specifying the categories

Specify the column or expression whose values determine the categories. You can select a column name from the dropdown listbox or type an expression:



There will be an entry along the Category axis for each different value of the column or expression you specify here.

### Using display values of data

If you are graphing columns that use code tables—data is stored with a data value but displayed to you with more meaningful display values—by default the graph will use the column's data values. To have the graph use a column's display values, use the `LookupDisplay` function when specifying Category or Series. `LookupDisplay` returns a string that matches the display value for a column:

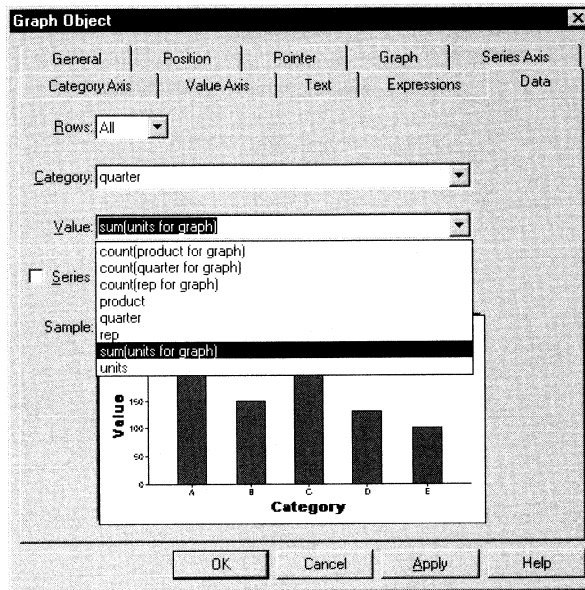
`LookupDisplay ( column )`

**FOR INFO** For more about code tables, see "Defining a code table" on page 288. For more about `LookupDisplay`, see `LookUpDisplay` on page 784.

## Specifying the values

InfoMaker populates the Value dropdown listbox in the Data property page. The listbox includes the names of all the retrieved columns as well as the following aggregate functions:

- ◆ Count for all non-numeric columns
- ◆ Sum for all numeric columns



Select an item from the dropdown listbox or type an expression. For example, if you want to graph the sum of units sold, you can specify:

```
sum(units for graph)
```

To graph 110 percent of the sum of units sold, you can specify:

```
sum(units*1.1 for graph)
```

## Specifying the series

Graphs can have one or more series. You specify this in the Data property page.

### Single-series graphs

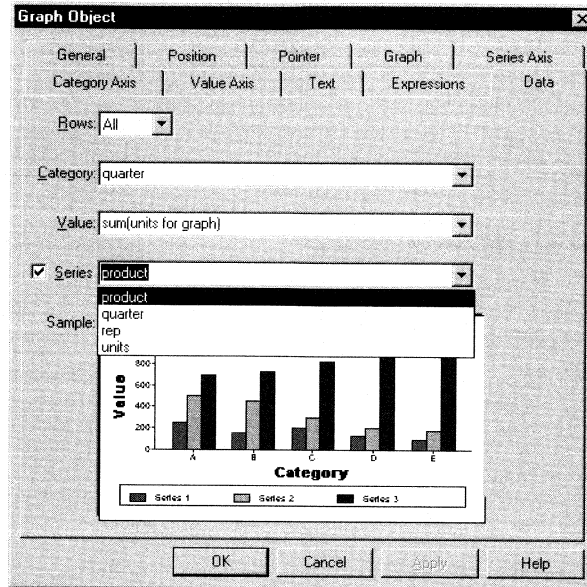
If you want only one series (that is, if you want to graph all retrieved rows as one series of values), leave the Series box empty.



## Multiple-series graphs

If you want to graph more than one series, select the Series checkbox and specify the column that will provide the series values.

You can select column names from the dropdown listbox:



There will be a set of data points for each different value of the column you specify here. For example, if you specify in the Series box a column that has 10 values, then your graph will have 10 series: one set of data points for each different value of the column.

## Using expressions

You can also specify expressions for Series. For example, you could specify the following for Series:

Units / 1000

In this case, if a table had unit values of 10,000, 20,000, and 30,000, the graph would show series values of 10, 20, and 30.

## Specifying multiple entries

You can specify more than one of the retrieved columns to serve as series. Separate multiple entries by commas.

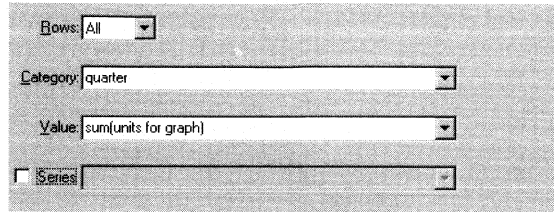
You need to specify the same number of entries in the Value box as you do in the Series box. The first value in the Value box corresponds to the first series identified in the Series box, the second value corresponds to the second series, and so on. The example about graphing actual and projected sales in "Examples" on page 406 illustrates this technique.

## Examples

This section shows how to specify the data for several different graphs of the data in the Printer table in the Powersoft Demo Database. The table records quarterly unit sales of three printers by three sales representatives:

<b>Rep</b>	<b>Quarter</b>	<b>Product</b>	<b>Units</b>
Simpson	Q1	Stellar	12
Jones	Q1	Stellar	18
Perez	Q1	Stellar	15
Simpson	Q1	Cosmic	33
Jones	Q1	Cosmic	5
Perez	Q1	Cosmic	26
Simpson	Q1	Galactic	6
Jones	Q1	Galactic	2
Perez	Q1	Galactic	1
...	...	...	...
Simpson	Q4	Stellar	30
Jones	Q4	Stellar	24
Perez	Q4	Stellar	36
Simpson	Q4	Cosmic	60
Jones	Q4	Cosmic	52
Perez	Q4	Cosmic	48
Simpson	Q4	Galactic	3
Jones	Q4	Galactic	3
Perez	Q4	Galactic	6

**Graphing total sales** Say you want to graph total sales of printers in each quarter. Retrieve all the columns into a report and create a graph with the following data specification:

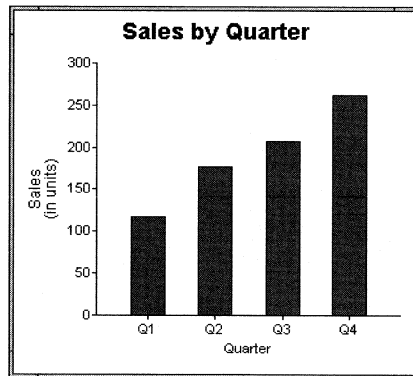


The screenshot shows a dialog box for configuring a graph. It has four fields:

- Rows:** A dropdown menu set to "All".
- Category:** A text box containing "quarter".
- Value:** A text box containing "sum(units for graph)".
- Series:** An empty text box.

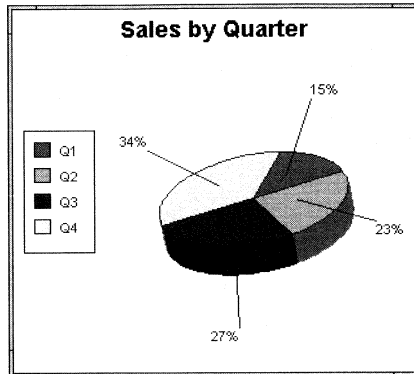
You want to graph sales by quarter, so the Quarter column serves as the category. Because the Quarter column has four values (Q1, Q2, Q3, and Q4), there will be four categories along the Category axis. You want only one series (total sales in each quarter), so you can leave the Series box empty as done in the preceding dialog box, or type a string literal to identify the series in a legend. You want to graph total sales in each quarter, so the Value box is specified as sum(units for graph).

Here is the resulting column graph. InfoMaker automatically generates the category text based on the data in the table:



In the preceding graph, there is one set of data points (one series) across four quarters (the category values).

The following is a pie graph, which has exactly the same properties as the column graph above except for the type, which is 3D Pie:



In pie graphs, categories are shown in the legend.

**Graphing unit sales of each printer** Say you want to graph total quarterly sales of each printer. Create a graph with the following data specification:

Rows: All

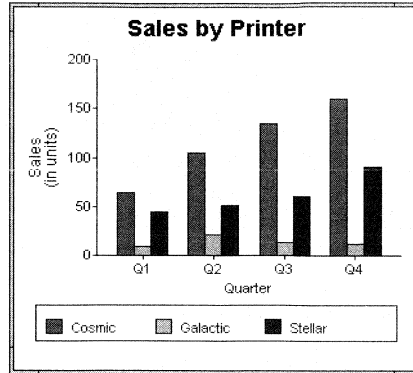
Category: quarter

Value: sum(units for graph)

Series: product

You want a different series for each printer, so the column Product serves as the series. Because the Product column has three values (Cosmic, Galactic, and Stellar), there will be three series in the graph. As in the first example, you want a value for each quarter, so the Quarter column serves as the category. And you want to graph total sales in each quarter, so the Value box is specified as sum(units for graph).

Here is the resulting graph. InfoMaker automatically generates the category and series labels based on the data in the table. The series labels display in the graph's legend:



**Graphing unit sales by representative** Say you want to graph the quarterly sales made by each representative. Here you want a different series for each representative, so the column Rep serves as the series. Your category is still the Quarter column. And you still want to graph total sales in each quarter for each series, so the Value box is still sum(units for graph):

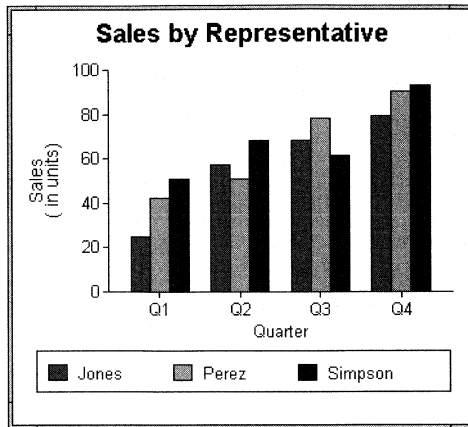
Rows: All

Category: quarter

Value: sum(units for graph)

Series: rep

Here is the resulting graph:



**Graphing unit sales by representative and total sales** Say you want to graph sales by representative, as shown above, plus total sales for each printer. Here is how to specify the data:

Rows: All

Category: quarter, "Total"

Value: sum(units for graph), sum(units for graph)

Series: rep, rep

Here you have two types of categories: the first is Quarter, which shows quarterly sales, as in the preceding graph. You also want a category for total sales. There is no corresponding column in the report, so you can simply type the literal "Total" to identify the category. The Category box looks like this:

```
quarter, "Total"
```

You separate multiple entries with a comma.

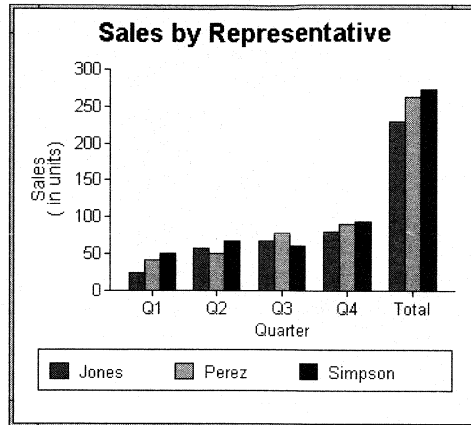
For each of these category types you want to graph the sum of units sold for each representative. So the Value entry is:

```
sum(units for graph), sum(units for graph)
```

And the Series entry is:

```
rep, rep
```

Here is the resulting graph:



Notice that InfoMaker uses the literal "Total" supplied in the Category box in the Graph Data window as a value in the Category axis.

**Graphing actual and projected sales** Say you want to graph total quarterly sales of all printers and also want to graph projected sales for next year. You figure that next year you will sell about 10 percent more printers. Here is how to specify the data:

Rows: All

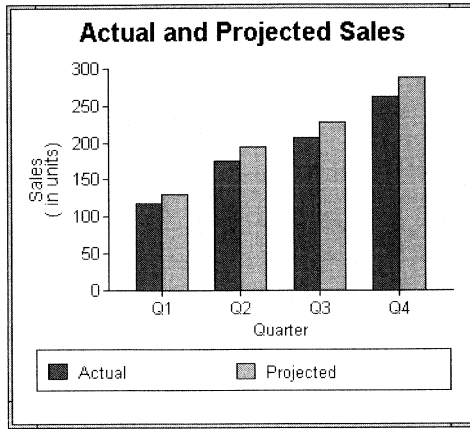
Category: quarter

Value:  $\text{sum}(\text{units for graph}), \text{sum}(\text{units} * 1.1 \text{ for graph})$

Series: 'Actual', 'Projected'

You are using labels to identify two series, Actual and Projected. Note the single quotation marks around the literals. For Values, you enter the expressions that correspond to Actual and Projected sales. For Actual, you use the same expression as in the examples above,  $\text{sum}(\text{units for graph})$ . For Projected sales, you multiply each unit sale by 1.1 to get the 10 percent increase. Therefore, the second expression is  $\text{sum}(\text{units} * 1.1 \text{ for graph})$ .

Here is the resulting graph. InfoMaker uses the literals you typed for the series as the series labels in the legend:



## Using overlays

It is often useful to call special attention to one of the series in a graph, particularly in a bar or column graph. You can do that by defining the series as an **overlay**. An overlay series is graphed as a line on top of the other series in the graph. To define a series as an overlay, define the series in the Graph Object Data property page as follows:

- ◆ If specifying a column name to identify the series, specify this for the series:

"@overlay~t" + *ColumnName*

- ◆ If using a label to identify the series, specify this for the series:

"@overlay~tSeriesLabel "

### Examples

Say you want to graph sales in each quarter and overlay the sales of each individual printer. Specify the graph's data like this:

Rows:

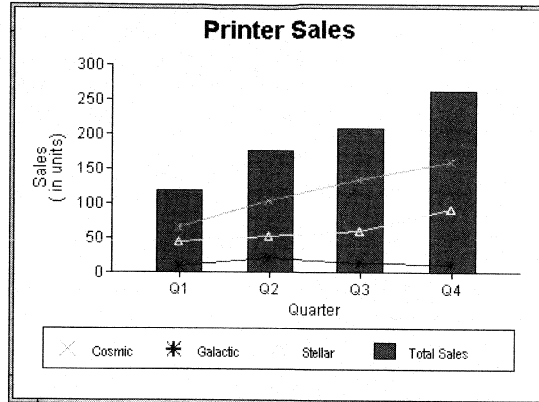
Category:

Value:

Series:



Here is the resulting graph:



To graph unit sales of printers by quarter and overlay the largest sale made in each quarter, specify the graph's data like this:

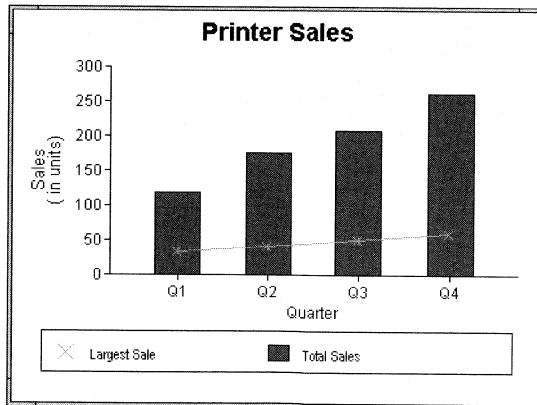
Rows: All

Category: quarter

Value: sum(units for graph), max(units for graph)

Series: "Total Sales", "@overlay~tLargest Sale"

Here is the resulting graph:



## Using the Graph presentation style

Instead of embedding a graph in a report, you can use the Graph presentation style to create a report that is only a graph—the underlying data is not displayed.

❖ **To use the Graph presentation style:**

- 1 Open the Report painter and select New in the Select Report dialog box. The New Report dialog box displays.
- 2 Select a data source and the Graph presentation style, then click OK. You are prompted to specify the data.
- 3 Specify the data you want retrieved into the report.  
**FOR INFO** For more information, see Chapter 4, "Defining Reports".  
The Report painter workspace displays, and you are prompted to specify the data for the graph in the graph's property sheet.
- 4 Enter the definitions for the series, categories, and values, as described in "Associating data with a graph" on page 401.  
Note that the Rows box is protected. When using the Graph presentation style, the graph always graphs all rows; you cannot specify page or group.
- 5 Click OK.  
A model of the graph displays in the entire workspace.
- 6 Specify the properties of the graph, as described in "Defining a graph's properties" next.
- 7 Save the report in a library.

## Defining a graph's properties

This section describes properties of a graph.

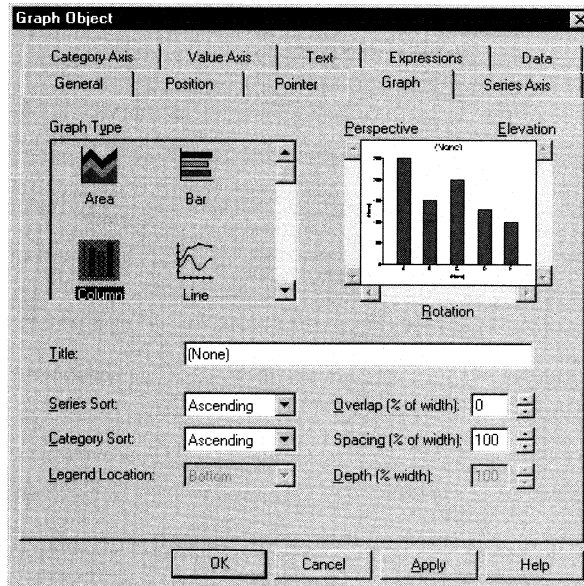
To define the properties of a graph, you use the graph's property sheet.

**FOR INFO** For general information about the ten tabbed property pages on the Graph Object property sheet, see "Using the graph's property sheet" on page 398.

## Using the Graph property page of the graph's property sheet

One of the first things you will probably do with a graph is name it and define its basic properties. You can do that on the Graph property page of the graph's property sheet.

- ❖ **To specify the basic properties of a graph:**
  - ◆ Select Properties from the graph's popup menu and then select the Graph tab:



About the model graph

As you modify your graph's properties, InfoMaker updates the model graph shown on the Graph property page so you can get an idea of the graph's basic layout:

- ◆ InfoMaker uses the graph title and axis labels you specify
- ◆ InfoMaker uses sample data (not data from your report) to illustrate series, categories, and values

## **Naming a graph**

Typically, you will not need to name a graph (the name is used to refer to the graph in PowerBuilder scripts).

❖ **To name a graph:**

- 1 Select Properties from the graph's popup menu and then select the General tab.
- 2 Assign a meaningful name to the graph in the Name box.
- 3 Click Apply or OK.

## **Defining a graph's title**

The title displays at the top of the graph.

❖ **To specify a graph's title:**

- 1 Select Properties from the graph's popup menu and then select the Graph tab.
- 2 Enter a title in the Title box.
- 3 Click Apply or OK.

---

### **Multiline titles**

You can force a new line in a title by embedding ~n.

---

**FOR INFO** For information about specifying properties for the title text, see "Specifying text properties for titles, labels, axes, and legends" on page 418.

## Specifying the type of graph

You can change the graph type anytime.

### ❖ To specify the graph type:

- 1 Select Properties from the graph's popup menu and then select the Graph tab.
- 2 Scroll the presentation of graph types to see all that are available.
- 3 Click the icon representing the graph type you want.
- 4 Click Apply or OK.

## Using legends

A legend provides a key to your graph's series.

### ❖ To include a legend for a series in a graph:

- 1 Select Properties from the graph's popup menu and then select the Graph tab.
- 2 Specify where you want the legend to appear by selecting a value in the Legend Location dropdown listbox.
- 3 Click Apply or OK.

**FOR INFO** For information on specifying text properties for the legend, see "Specifying text properties for titles, labels, axes, and legends" on page 418.

## Sorting data

You can specify how to sort the data for series and categories. By default, the data is sorted in ascending order.

### ❖ To specify how to sort the data for series and categories in a graph:

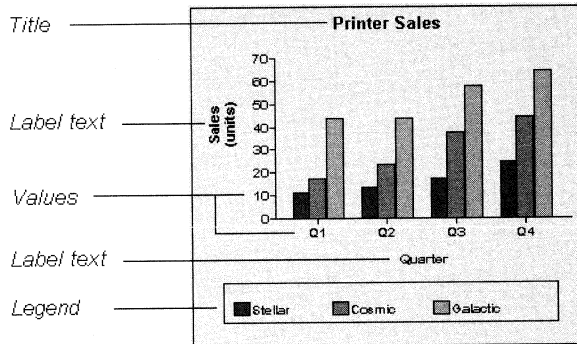
- 1 Select Properties from the graph's popup menu and then select the Graph tab.
- 2 Select Ascending (order), Descending (order), or Unsorted in the Series Sort box or the Category Sort box.
- 3 Click Apply or OK.

You can choose unsorted, sort in ascending order, or sort in descending order.

## Specifying text properties for titles, labels, axes, and legends

A graph can have four text elements:

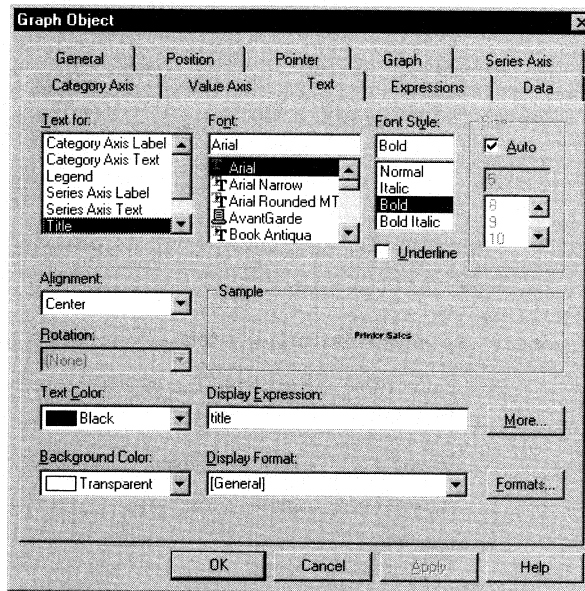
- ◆ Title
- ◆ Labels for the axes
- ◆ Text that shows the values along the axes
- ◆ Legend



You can specify properties for each text element.

- ❖ **To specify text properties for the title, labels, axes values, and legend of a graph:**
  - 1 Select Properties from the graph's popup menu and then select the Text tab.

- 2 Select a text element from the list in the Text For box:



- 3 Specify the font and its characteristics.
- 4 Click Apply or OK.

## Using Auto Size

With Auto Size in effect, InfoMaker resizes the text appropriately whenever the graph is resized. With Auto Size disabled, you specify the font size of a text element explicitly.

- ❖ **To have InfoMaker automatically size a text element in a graph:**
  - 1 Select Properties from the graph's popup menu and then select the Text tab.
  - 2 Select a text element from the list in the Text For box.
  - 3 Select the Auto checkbox (this is the default).
  - 4 Click Apply or OK.
- ❖ **To specify a font size for a text element in a graph:**
  - 1 Select Properties from the graph's popup menu and then select the Text tab.

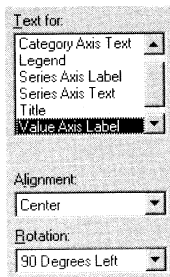
- 2 Select a text element from the list in the Text For box.
- 3 Deselect the Auto checkbox.
- 4 Select the Font size in the Size dropdown listbox.
- 5 Click Apply or OK.

## Rotating text

For all the text elements, you can specify the number of degrees by which you want to rotate the text.

### ❖ To specify rotation for a text element in a graph:

- 1 Select Properties from the graph's popup menu and then select the Text tab.
- 2 Select a text element from the list in the Text For box.
- 3 Specify the rotation you want:



- 4 Click Apply or OK.

Changes you make here are shown in the model graph in the workspace and on the Graph property page of the graph's property sheet.

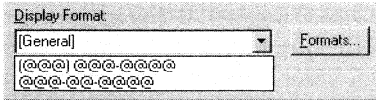
## Using display formats

### ❖ To use a display format for a text element in a graph:

- 1 Select Properties from the graph's popup menu and then select the Text tab.
- 2 Select a text element from the list in the Text For box.



- Choose an existing display format for the text from the Display Format dropdown listbox, or define a new display format by clicking the Formats button:



- Click Apply or OK.

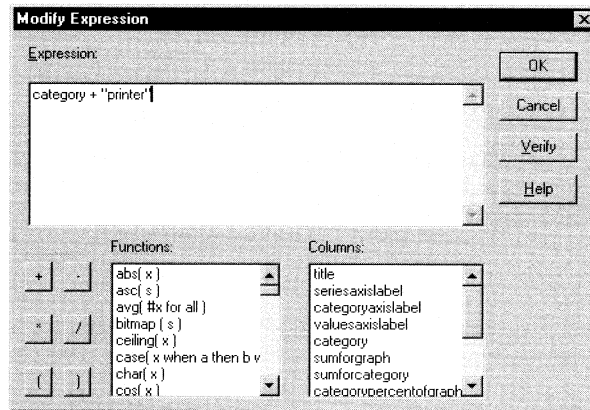
## Modifying display expressions

You can specify an expression for the text that is used for each graph element. The expression is evaluated at execution time.

### ❖ To specify an expression for a text element in a graph:

- Select Properties from the graph's popup menu and then select the Text tab.
- Select a text element from the list in the Text For box.
- Click the More button.

The Modify Expression dialog box displays:



- Specify the expression.

You can paste functions, column names, and operators. Included with column names in the Columns box are statistics about the columns, such as counts and sums.

- Click OK to return to the graph's property sheet.

6 Click Apply or OK.

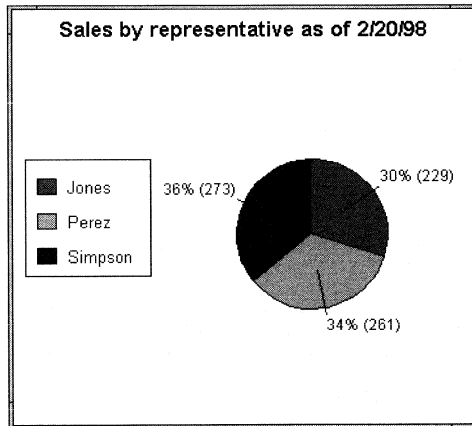
Example

Here's an example of using expressions to enhance a graph.

By default, when you generate a pie graph, InfoMaker puts the title at the top and labels each slice of the pie with the percentage each slice is of the whole. Percentages are accurate to two decimal places.

The following graph has been enhanced as follows:

- ◆ The current date displays in the title
- ◆ The percentages are rounded to integers
- ◆ The raw data for each slice is shown in addition to the percentages



To accomplish this, the display expressions were modified for the title and pie graph labels:

Text element	Original expression	Modified expression
Title	title	title + " as of " + date(today())
Pie graph labels	if(seriescount > 1, series, string(percentofseries, "0.00%"))	if(seriescount > 1, series, string(percentofseries, "0%") + (" + value + "))

## Specifying overlap and spacing

With bar and column charts, you can specify the following properties:

Property	Meaning
Overlap	The percentage by which bars or columns overlap each other. The default is 0 percent, meaning no overlap
Spacing	The amount of space to leave between bars or columns. The default is 100 percent, which leaves a space equal to the width of a bar or column

❖ **To specify overlap and spacing for the bars or columns in a graph:**

- 1 Select Properties from the graph's popup menu and then select the Graph tab.
- 2 Specify a percentage for Overlap (% of width) and Spacing (% of width).
- 3 Click Apply or OK.

## Specifying axis properties

Graphs have two or three axes. You specify the axes' properties from the Category Axis, Value Axis, or Series Axis property pages of the graph's property sheet.

❖ **To specify properties for an axis of a graph:**

- 1 Select Properties from the graph's popup menu.
- 2 Select the Category Axis tab, the Value Axis tab, or the Series Axis tab.  
If you are not working with a 3D graph, the Series Axis options are disabled.
- 3 Specify the properties as described next.
- 4 Click Apply or OK.

## Specifying text properties

You can specify the characteristics of text that displays for each axis. There are two kinds of text associated with an axis:

Type of text	Meaning
Text	Text that identifies the values for an axis
Label	Text that describes the axis. You specify the label text in a painter. You can use ~n to embed a new line within a label

FOR INFO For information on specifying properties for the text, see "Specifying text properties for titles, labels, axes, and legends" on page 418.

## Specifying data types

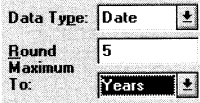
The data graphed along the Value, Category, and Series axes has an assigned data type. The Series axis always has the data type String. The Value and Category axes can have the following data types:

Axis	Possible data types
Value	Number, Date, DateTime, Time
Category (for scatter graph)	Number, Date, DateTime, Time
Category (other graph types)	String, Number, Date, DateTime, Time

InfoMaker automatically assigns the data types based on the data type of the corresponding column; you do not specify them.

## Scaling axes

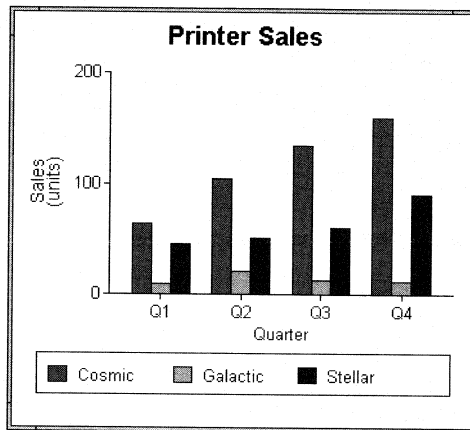
You can specify several properties that define the scaling used along numeric axes:

Property	Meaning
Autoscale	If selected (the default), InfoMaker automatically assigns a scaling for the numbers along the axis
Round Maximum To	<p>Specifies how to round the end points of the axis (note that this just rounds the range displayed along the axis; it doesn't round the data itself)</p> <p>You can specify a number and a unit. The unit is based on the data type; you can specify Default as the unit to have InfoMaker decide for you</p> <p>For example, if the Value axis is a Date column, you can specify that you want to round the end points of the axis to the nearest five years. In this case, if the largest data value is the year 1993, the axis will extend up to 1995, which is 1993 rounded to the next highest five-year interval:</p> 

Property	Meaning
Minimum, Maximum	The smallest and largest numbers to appear on the axis (disabled if you have selected Autoscale)
Scale	Specifies linear or logarithmic scaling (common or natural)

## Using major and minor divisions

You can divide axes into divisions. Each division is identified by a tick mark, which is a short line that intersects an axis. The following graph's Value axis is divided into two major divisions. One goes from 0 to 100. The other goes from 100 to 200:

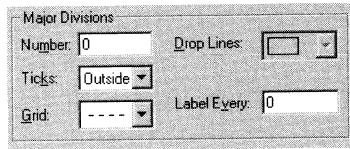


By default, InfoMaker divides the axes automatically into major divisions.

❖ **To define divisions for an axis of a graph:**

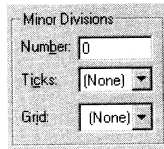
- 1 To divide an axis into a specific number of major divisions, type the number of divisions you want in the Number box in the Major Divisions group.

Leave the number 0 to have InfoMaker automatically create divisions.



By default, InfoMaker labels each tick mark in major divisions. If you don't want each tick mark labeled, enter a value in the Label Every box. For example, if you enter 2, InfoMaker will label every second tick mark for the major divisions.

- 2 To use minor divisions, which are divisions within each major division, type the appropriate number in the Number box in the Minor Divisions group. To not use minor divisions, leave the number 0:




---

**When using logarithmic axes**

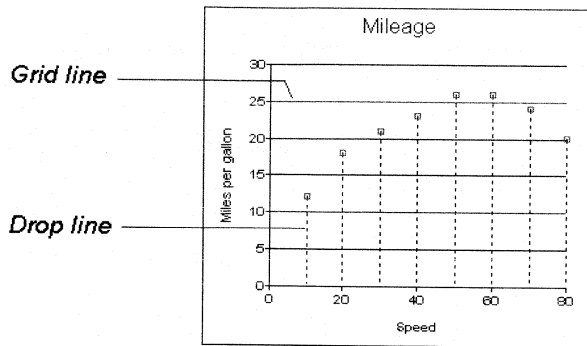
If you want minor divisions, specify 1; otherwise, specify 0.

---

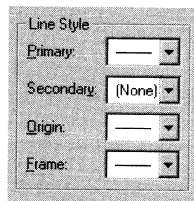
Specifying division lines

You can specify lines to represent the divisions:

Line	Meaning
Grid line	A line that extends from a tick mark across the graph. Grid lines make graphs easier to read
Drop line	A line that extends vertically from a data point to its axis (not available for all graph types)



## Using line styles



You can define line styles for the following components of an axis:

Component	Meaning
Primary	The axis itself
Secondary	The axis parallel to and opposite the primary axis
Origin	A grid line that represents the value zero
Frame	The frame for the axis in 3D graphs (disabled for 2D graphs)

## Specifying a border

You can specify the border that InfoMaker places around a graph.

### ❖ To specify a border for a graph:

- 1 Select the graph.
- 2 Select the type of border to use from the Border dropdown toolbar.

## Specifying a pointer

You can specify a pointer to use when the mouse is over a graph while displaying the report.

❖ **To specify a pointer for a graph:**

- 1 Select Properties from the graph's popup menu and then select the Pointer tab.
- 2 Select a stock pointer from the list, or select a CUR file containing a pointer.
- 3 Click Apply or OK.

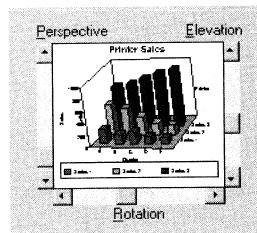
## Specifying point of view in 3D graphs

If you are defining a 3D graph, you can specify the point of view that InfoMaker uses when displaying the graph.

❖ **To specify a 3D graph's point of view:**

- 1 Select Properties from the graph's popup menu and then select the Graph tab.

The Graph property page includes a representation of the graph and scroll bars that let you change the point of view for a 3D graph:



- 2 Adjust the point of view along the three dimensions of the graph:
  - ◆ To change the perspective, move the scroll box in the left scrollbar.
  - ◆ To rotate the graph, move the scroll box in the bottom scrollbar.
  - ◆ To change the elevation, move the scroll box in the right scrollbar.
- 3 Define the depth of the graph (the percent the depth is of the width of the graph) by entering a number in the Depth (% width) box or using the scrollbars on the graph representation to increase or decrease the depth.



4 Click OK.



About this chapter

This chapter describes how to build crosstabs.

Contents

<b>Topic</b>	<b>Page</b>
About crosstabs	432
Creating crosstabs	436
Associating data with a crosstab	437
Previewing crosstabs	443
Enhancing crosstabs	444

# About crosstabs

Cross tabulation is a useful technique for analyzing data. By presenting data in a spreadsheet-like grid, a crosstab lets you view summary data instead of a long series of rows and columns. For example, in a sales report you might want to summarize the quarterly unit sales of each product.

In InfoMaker, you create crosstabs by using the Crosstab report presentation style. When data is retrieved into the report, the crosstab processes all the data and presents the summary information that you have defined for it.

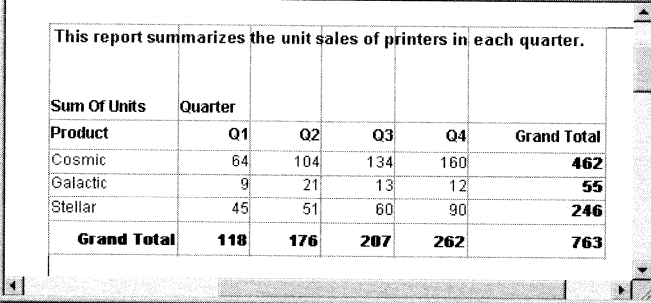
### An example

Crosstabs are easiest to understand through an example. Consider the Printer table in the Powersoft Demo Database. It records quarterly unit sales of printers made by sales representatives in one year (this is the same data that was used to illustrate graphs in the preceding chapter):

<b>Rep</b>	<b>Quarter</b>	<b>Product</b>	<b>Units</b>
Simpson	Q1	Stellar	12
Jones	Q1	Stellar	18
Perez	Q1	Stellar	15
Simpson	Q1	Cosmic	33
Jones	Q1	Cosmic	5
Perez	Q1	Cosmic	26
Simpson	Q1	Galactic	6
Jones	Q1	Galactic	2
Perez	Q1	Galactic	1
.	.	.	.
.	.	.	.
.	.	.	.
Simpson	Q4	Stellar	30
Jones	Q4	Stellar	24
Perez	Q4	Stellar	36
Simpson	Q4	Cosmic	60
Jones	Q4	Cosmic	52
Perez	Q4	Cosmic	48

Rep	Quarter	Product	Units
Simpson	Q4	Galactic	3
Jones	Q4	Galactic	3
Perez	Q4	Galactic	6

This information can be summarized in a crosstab. Here is a crosstab that shows unit sales by printer for each quarter:



Sum Of Units	Quarter					
Product	Q1	Q2	Q3	Q4	Grand Total	
Cosmic	64	104	134	160	<b>462</b>	
Galactic	9	21	13	12	<b>55</b>	
Stellar	45	51	60	90	<b>246</b>	
<b>Grand Total</b>	<b>118</b>	<b>176</b>	<b>207</b>	<b>262</b>	<b>763</b>	

The first-quarter sales of Cosmic printers displays in the first data cell (as you can see from the data in the Printer table shown before the crosstab, in Q1 Simpson sold 33 units, Jones sold 5 units, and Perez sold 26 units—totaling 64 units). InfoMaker calculates each of the other data cells the same way.

To create this crosstab, all you have to do is tell InfoMaker which database columns contain the raw data for the crosstab and InfoMaker does all the data summarization automatically.

#### What crosstabs do

Crosstabs perform 2-dimensional analysis:

- ◆ The first dimension is displayed as columns across the crosstab.

In the preceding crosstab, it is the quarter, whose values are in the Quarter column in the database table.

- ◆ The second dimension is displayed as rows down the crosstab.

In the preceding crosstab, it is the type of printer, whose values are in the Product column in the database table.

Each cell in a crosstab is the intersection of a column (the first dimension) and a row (the second dimension). The numbers that appear in the cells are calculations based on both dimensions. In the preceding crosstab, it is the sum of unit sales for the quarter in the corresponding column and printer in the corresponding row.

Crosstabs can also include summary statistics. The preceding crosstab totals the sales for each quarter in the last row and the total sales for each printer in the last column.

How crosstabs are implemented in InfoMaker

Crosstabs in InfoMaker are implemented as grid reports. Because crosstabs are grid reports, you can resize and reorder columns when you run the crosstab.

---

### Running a crosstab

You can run a crosstab by previewing it in the Report painter and by running it from an executable file.

---

## Two types of crosstabs

There are two types of crosstabs:

- ◆ Dynamic
- ◆ Static

Dynamic crosstabs

With **dynamic crosstabs**, InfoMaker builds all the columns and rows in the crosstab dynamically when you run the crosstab. The number of columns and rows in the crosstab match the data that exists at execution time.

Using the preceding crosstab as an example, if a new printer was added to the database after the crosstab was saved, there would be an additional row in the crosstab when it is run. Similarly, if one of the quarter's results was deleted from the database after the crosstab was saved, there would be one less column in the crosstab when it is run.

By default, crosstabs you build are dynamic.

Static crosstabs

**Static crosstabs** are quite different.

With static crosstabs, InfoMaker establishes the columns in the crosstab based on the data in the database when you *define* the crosstab (it does this by retrieving data from the database when you initially define the crosstab in the Report painter workspace). No matter what values are in the database when you later run the crosstab, the crosstab will always have the same columns as when you defined it.

Using the preceding crosstab as an example, if there were four quarters in the database when you defined and saved the crosstab, there would always be four columns (Q1, Q2, Q3, and Q4) in the crosstab at execution time, even if the number of columns changed in the database.

Advantages of  
dynamic crosstabs

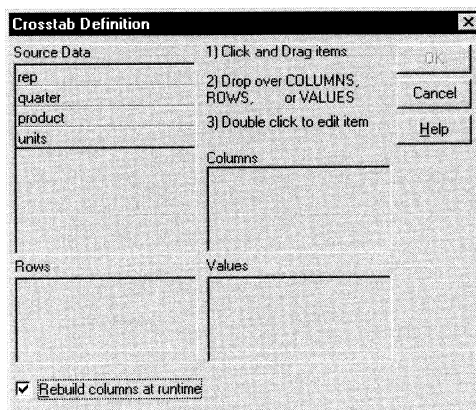
In most cases, you will want to use dynamic crosstabs, for the following reasons:

- ◆ You can define dynamic crosstabs very quickly because no database access is required at definition time.
- ◆ Dynamic crosstabs always use the current data to build the columns and rows in the crosstab. Static crosstabs show a snapshot of columns as they were when the crosstab was defined.
- ◆ Dynamic crosstabs are easy to modify: all properties for the dynamically built columns are replicated during execution automatically. With static crosstabs, you must work with one column at a time.

## Creating crosstabs

❖ **To create a crosstab:**

- 1 Open the Report painter and select New in the Select Report dialog box. The New Report dialog box displays.
- 2 Select a data source and the Crosstab presentation style, then click OK. You are prompted to specify the data.
- 3 Specify the data you want retrieved into the report.  
 FOR INFO For more information, see Chapter 4, "Defining Reports".  
 You are prompted to specify the data for the columns, rows, and cell values in the crosstab:



- 4 Enter the definitions for the columns, rows, and cell values in the crosstab.  
 FOR INFO See "Associating data with a crosstab" on page 437.
- 5 Click OK.  
 InfoMaker places the crosstab in the workspace.
- 6 (Optional) Preview the crosstab to see how it looks.  
 FOR INFO See "Previewing crosstabs" on page 443.
- 7 (Optional) Specify other properties of the crosstab.  
 FOR INFO See "Enhancing crosstabs" on page 444.
- 8 Save the report in a library.



## Associating data with a crosstab

You associate crosstab columns, rows, and cell values with columns in a database table or other data source.

### ❖ To associate data with a crosstab:

- 1 If you are defining a new crosstab, the Crosstab Definition dialog box displays after you specify the data source.
- 2 Specify the database columns that will populate the columns, rows, and values in the crosstab, as described below.
- 3 To build a dynamic crosstab, make sure the Rebuild Columns At Runtime box is selected.

**FOR INFO** For information about static crosstabs, see "Creating static crosstabs" on page 453.

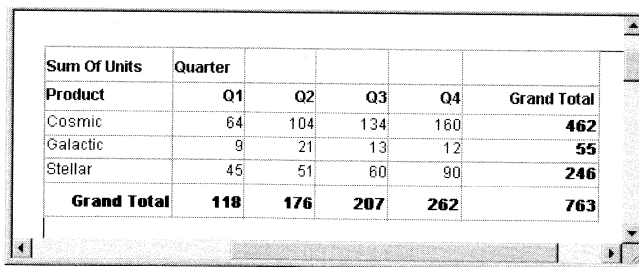
- 4 Click OK.

## Specifying the information

To define the crosstab, you simply drag the column names from the Source Data box in the Crosstab Definition dialog box into the Columns, Rows, or Values box, as appropriate.

If you change your mind or want to edit the report later, drag the column name out of the Columns, Row, or Values box and drop it. Then specify a different column.

The process is illustrated using the following dynamic crosstab. The columns in the database are Rep, Quarter, Product, and Units. The crosstab shows the number of printers sold by Quarter:



Sum Of Units	Quarter				
Product	Q1	Q2	Q3	Q4	Grand Total
Cosmic	64	104	134	160	<b>462</b>
Galactic	9	21	13	12	<b>55</b>
Stellar	45	51	60	90	<b>246</b>
<b>Grand Total</b>	<b>118</b>	<b>176</b>	<b>207</b>	<b>262</b>	<b>763</b>

## Specifying the columns

You use the Columns box to specify one or more of the retrieved columns to provide the columns in the crosstab. When you run the crosstab, there will be one column in the crosstab for each unique value of the database column(s) you specify here.

❖ **To specify the crosstab's columns:**

- ◆ Drag the database column from the Source Data box into the Columns box.

Using the printer example, to create a crosstab where the quarters form the columns, specify Quarter as the Columns value. Because there are four values in the table for Quarter (Q1, Q2, Q3, and Q4), there will be four columns in the crosstab.

## Specifying the rows

You use the Rows box to specify one or more of the retrieved columns to provide the rows in the crosstab. When you run the crosstab, there will be one row in the crosstab for each unique value of the database column(s) you specify here.

❖ **To specify the crosstab's rows:**

- ◆ Drag the database column from the Source Data box into the Rows box.

Using the printer example, to create a crosstab where the printers form the rows, specify Product as the Rows value. Because there are three products (Cosmic, Galactic, and Stellar), during execution there will be three rows in the crosstab.

---

### **Display values are used**

If you specify columns in the database that use code tables—data is stored with a data value but displayed with more meaningful display values—the crosstab uses the column's display values, not the data values.

**FOR INFO** For more about code tables, see Chapter 6, "Displaying and Validating Data".

---

## Using expressions

Instead of simply specifying database columns, you can use any valid InfoMaker expression to define the columns, rows, and values used in the crosstab. You can use any InfoMaker function in the expression.

For example, say a table contains a date column named `SaleDate`, and you want a column in the crosstab for each month. You could enter the following expression for the Columns definition:

```
Month(SaleDate)
```

The `Month` function returns the integer value (1–12) for the specified month. Using this expression, you get columns labeled 1 through 12 in the crosstab. Each database row for January sales is evaluated in the column under 1, each database row for February sales is evaluated in the column under 2, and so on.

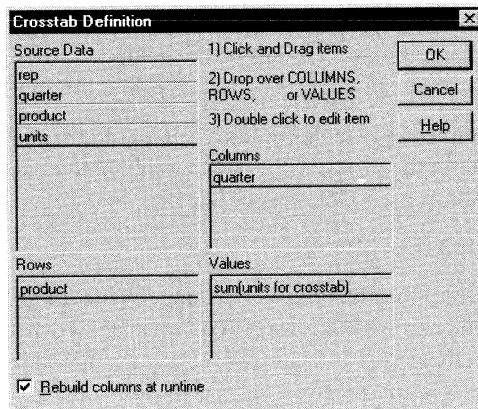
### ❖ To specify an expression for columns, rows, or values:

- 1 In the Crosstab Definition dialog box, double-click the item in the Columns, Rows, or Values box.  
The Modify Expression dialog box displays.
- 2 Specify the expression and click OK.

## What happens

After you have specified the data for the crosstab's columns, rows, and values, InfoMaker displays the crosstab definition in the Report painter workspace. The crosstab is implemented as a grid report.

To create the dynamic crosstab shown earlier, you would enter these values:



In the workspace, the crosstab looks like this:

Sum Of Units	Quarter		
<b>Header [1] ↑</b>			
Product	@quarter	Grand Total	
<b>Header [2] ↑</b>			
product	units	crosstabsum(1)	
<b>Detail ↑</b>			
"Grand Total"	sum(units	sum(grand_sum_units for	
<b>Summary ↑</b>			
<b>Footer ↑</b>			

Notice that in the workspace InfoMaker shows the Quarter entries using the symbolic notation **@quarter** (with dynamic crosstabs, the actual data values are not known at definition time). **@quarter** is resolved into the actual data values (in this case, Q1, Q2, Q3, and Q4) when the crosstab runs.

The crosstab is generated with summary statistics: the rows and columns are totaled for you.

At this point, you can run the crosstab by clicking the Preview button:

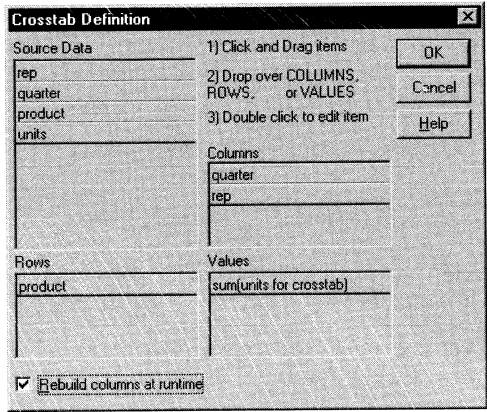
Sum Of Units	Quarter				
Product	Q1	Q2	Q3	Q4	Grand Total
Cosmic	64	104	134	160	462
Galactic	9	21	13	12	55
Stellar	45	51	60	90	246
<b>Grand Total</b>	<b>118</b>	<b>176</b>	<b>207</b>	<b>262</b>	<b>763</b>

- ◆ Because Quarter was selected as the Columns definition, there is one column in the crosstab for each unique quarter (Q1, Q2, Q3, and Q4)
- ◆ Because Product was selected as the Rows definition, there is one row in the crosstab for each unique product (Cosmic, Galactic, and Stellar)
- ◆ Because sum(units for crosstab) was selected as the Values definition, each cell contains the total unit sales for the corresponding quarter (the Columns definition) and product (the Rows definition)
- ◆ InfoMaker displays the grand totals for each column and row in the crosstab

## Specifying more than one row or column

Typically you will specify one database column as the Columns definition and one database column for the Rows definition, as in the printer crosstab. But you can specify as many columns (or expressions) as you want.

For example, consider the following, which specifies two database columns as the Columns definition:



Here we want columns in the crosstab for quarters and for sales reps.

InfoMaker displays this in the workspace:

Sum Of Units	Quarter	Rep	
<b>Header [1]</b>			
	@quarter	@quarter	Sum Of Units
<b>Header [2]</b>			
Product	@rep		Grand Total
<b>Header [3]</b>			
product	units	crosstabsum(1, 2, "@quarter crosstabsum(1)")	
<b>Detail</b>			
"Grand Total"	sum(units sum(sum_units for all))	sum(grand_sum_units for	
<b>Summary</b>			
<b>Footer</b>			

This is what you get during execution:

Sum Of Units	Quarter			Rep					
	Q1				Q1 Sum Of Units			Q2	Q2 Sum Of Units
Product	Jones	Perez	Simpson		Jones	Perez	Simpson		
Cosmic	5	26	33		64	36	28	40	104
Galactic	2	1	6		9	6	3	12	21
Stellar	18	15	12		45	15	20	16	51
<b>Grand Total</b>	<b>25</b>	<b>42</b>	<b>51</b>		<b>118</b>	<b>57</b>	<b>51</b>	<b>68</b>	<b>176</b>

For each quarter, the crosstab shows sales of each printer by each sales representative.

## Previewing crosstabs

Once you have defined the data for the crosstab, you can run it to see the data. You run it by previewing the report.

❖ **To preview the crosstab:**

- 1 Click the Preview button.

*or*

Select Design>Preview from the menu bar.

You are now in preview. The bars that indicate the report bands disappear. InfoMaker retrieves the rows and performs the cross tabulation on the data.

- 2 Continue previewing your report.

You can do the same things you do when previewing standard grid reports. For example, you can resize columns, filter rows, sort rows, save data in an external file, and print the results.

**FOR INFO** For more on what you can do in preview, see Chapter 5, "Enhancing Reports".

- 3 When you have finished previewing the crosstab, click the Preview button.

You return to the workspace. Changes you made (such as resizing columns) are retained.

## Enhancing crosstabs

Once you have provided the data definitions, the crosstab is functional. But you will probably want to enhance it before using it. Because a crosstab is a grid report, you can enhance a crosstab using the same techniques you use in other reports. For example, you might want to:

- ◆ Sort or filter rows
- ◆ Change the column headers
- ◆ Specify fonts and borders
- ◆ Specify column display formats

**FOR INFO** For more on these and the other standard enhancements you can make to reports, see Chapter 5, "Enhancing Reports".

The rest of this section covers topics either unique to crosstabs or especially important when working with crosstabs:

- ◆ "Specifying basic properties" next
- ◆ "Modifying the data associated with the crosstab" on page 445
- ◆ "Changing the names used for the columns and rows" on page 446
- ◆ "Defining summary statistics" on page 447
- ◆ "Cross-tabulating ranges of values" on page 450
- ◆ "Creating static crosstabs" on page 453
- ◆ "Using property conditional expressions" on page 454

## Specifying basic properties

Crosstabs are implemented as grid reports, so you can specify the following grid properties for a crosstab:

- ◆ When grid lines are displayed
- ◆ What is allowed when you run the report
- ❖ **To specify the crosstab's basic properties:**
  - 1 Position the mouse below the footer band in the workspace and display the popup menu.
  - 2 Select Properties and then the General tab.



- 3 Specify basic crosstab properties in the Grid box on the General property page:

Option	Result
Display	<p><i>On</i> — Grid lines always display</p> <p><i>Off</i> — Grid lines never display (columns cannot be resized during execution)</p> <p><i>Display Only</i> — Grid lines display only when the crosstab displays online</p> <p><i>Print Only</i> — Grid lines display only when the contents of the crosstab are printed</p>
Column Moving	Columns can be moved during execution
Mouse Selection	Data can be selected during execution (and, for example, copied to the clipboard)
Row Resize	Rows can be resized during execution

## Modifying the data associated with the crosstab

When you initially define the crosstab, you associate the crosstab rows and columns with columns in a database table or other data source. You can change the associated data anytime in the Crosstab Definition dialog box.

- ❖ **To open the Crosstab Definition dialog box:**
  - 1 Position the mouse below the footer band in the workspace and display the popup menu.
  - 2 Select Crosstab from the popup menu.  
The Crosstab Definition dialog box displays.
- ❖ **To modify the data associated with a crosstab:**
  - 1 In the Crosstab Definition dialog box, fill in the boxes for Columns, Rows, and Values as described in "Associating data with a crosstab" on page 437.
  - 2 Click OK.

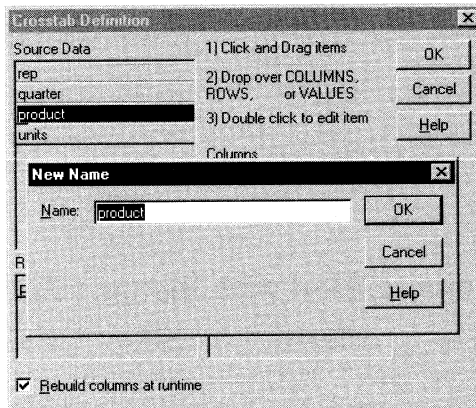
## Changing the names used for the columns and rows

Sometimes names of columns in the database are not very user friendly and may not be meaningful. You can change the names that are used to label rows and columns in crosstabs so that the data is easier to understand.

❖ **To change the names used in crosstabs:**

- 1 In the Crosstab Definition dialog box, double-click the name of the column in the Source Data box.

The New Name dialog box displays:



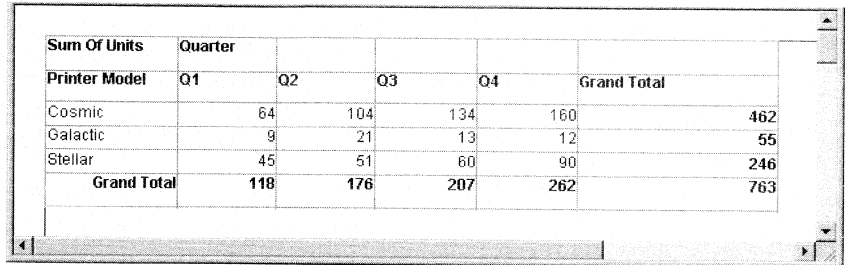
- 2 Specify the name you want to be used to label the corresponding column. You can have multiple-word labels by using underscores: underscores are replaced by spaces in the workspace and during execution.
- 3 Click OK.

InfoMaker changes the column name in the Source Data box and anywhere else the column is used.

**Example**

For example, if you want the Product column to be labeled *Printer Model*, double-click Product in the Crosstab Definition dialog box and specify **printer\_model** in the New Name dialog box.

When the crosstab runs, you will see this:



Sum Of Units	Quarter				
Printer Model	Q1	Q2	Q3	Q4	Grand Total
Cosmic	64	104	134	160	462
Galactic	9	21	13	12	55
Stellar	45	51	60	90	246
<b>Grand Total</b>	<b>118</b>	<b>176</b>	<b>207</b>	<b>262</b>	<b>763</b>

## Defining summary statistics

When you generate a crosstab, the columns and rows are automatically totaled for you. You can include other statistical summaries in crosstabs as well. To do that, you place computed fields in the workspace.

### ❖ To define a column summary:

- 1 Enlarge the summary band to make room for the summaries.
- 2 Click the Compute button in the PainterBar.  
*or*  
Select Objects>Computed Field from the menu bar.
- 3 Click the cell in the summary band where you want the summary to display.

The Computed Object property sheet displays.

- 4 Define the computed field.

For example, if you want the average value for a column specify **avg(units for all)**, where Units is the column providing the values in the crosstab.

Here is a crosstab that has been enhanced to show averages and maximum values for each column. This is the workspace:

This report summarizes the unit sales of printers in each quarter.

Sum Of Units	Quarter				
<b>Header [1] ↑</b>					
Product	@quarter	Grand Total			
<b>Header [2] ↑</b>					
product	units	<b>crosstabsum(1)</b>			
<b>Detail ↑</b>					
<b>"Grand Total"</b>	<b>sum(unitsum(grand_sum</b>				
	Average avg( units				
	Maximum max( unit:				
<b>Summary ↑</b>					
<b>Footer ↑</b>					

And this is the crosstab during execution:

This report summarizes the unit sales of printers in each quarter.

Sum Of Units	Quarter					
Product	Q1	Q2	Q3	Q4	Grand Total	
Cosmic	64	104	134	160	<b>462</b>	
Galactic	9	21	13	12	<b>55</b>	
Stellar	45	51	60	90	<b>246</b>	
<b>Grand Total</b>	<b>118</b>	<b>176</b>	<b>207</b>	<b>262</b>	<b>763</b>	
<b>Average</b>	<b>39</b>	<b>59</b>	<b>69</b>	<b>87</b>		
<b>Maximum</b>	<b>64</b>	<b>104</b>	<b>134</b>	<b>160</b>		

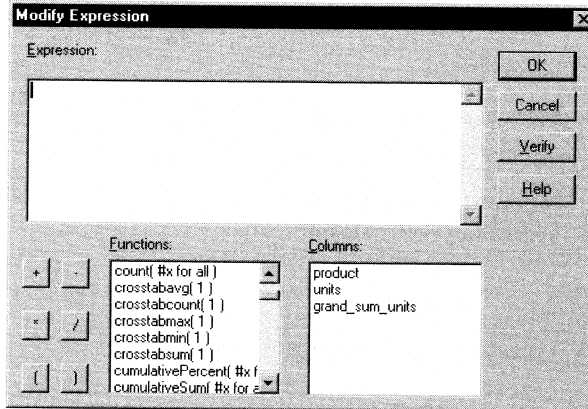
❖ **To define a row summary:**

- 1 Click the Compute button in the PainterBar.  
*or*  
Select Objects>Computed Field from the menu bar.
- 2 Click the empty cell to the right of the last column in the detail band.  
The Computed Object property sheet displays.
- 3 Define the computed field. You should use one of the crosstab functions, described next. To display the Modify Expression dialog box, click More.

## Using crosstab functions

There are five special functions you can use only in crosstabs: CrosstabAvg, CrosstabCount, CrosstabMax, CrosstabMin, and CrosstabSum.

These functions are listed in the Functions box when you define a computed field in a crosstab:



Each of these functions returns the corresponding statistic about a row in the crosstab (average, count, maximum value, minimum value, or sum). You place computed fields using these functions in the detail band in the workspace.

By default, InfoMaker places CrosstabSum in the detail band, which returns the total for the corresponding row.

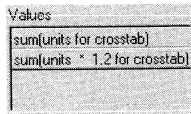
### How to specify the functions

Each of these functions takes one numeric argument, which refers to the expression defined for Values in the Crosstab Definition dialog box. The first expression for Values is numbered 1, the second is numbered 2, and so on.

Generally, crosstabs have only one expression for Values, so the argument for the crosstab functions is 1. So, for example, if you defined sum(units for crosstab) as your Values expression, InfoMaker places CrosstabSum(1) in the detail band.

But assume you want to cross-tabulate total unit sales and also a projection for future sales, assuming a 20 percent increase in sales (that is, sales that are 1.2 times the actual sales).

You would define two expressions for Values:



Here CrosstabSum(1) returns the total of sum(units for crosstab) for the corresponding row. CrosstabSum(2) returns the total for sum(units \* 1.2 for crosstab).

For more information

For complete information about defining computed fields, see Chapter 5, "Enhancing Reports".

For more about the crosstab functions, see Chapter 23, "DataWindow Painter and InfoMaker Functions".

## Cross-tabulating ranges of values

Often you want to build a crosstab where each row tabulates a *range* of values, instead of one discrete value. Similarly, you might want each column in the crosstab to correspond to a range of values.

For example, in cross-tabulating departmental salary information, you might want one row in the crosstab to count all employees making between \$30,000 and \$40,000, the next row to count all employees making between \$40,000 and \$50,000, and so on.

### ❖ To cross-tabulate ranges of values:

- 1 Determine the expression that results in the raw values being converted into one of a small set of fixed values.

Each of those values will form a row or column in the crosstab.

- 2 Specify the expression in the Columns or Rows box in the Crosstab Definition dialog box. You choose the box depending on whether you want the columns or rows to correspond to the range of values.
- 3 In the Values column, apply the appropriate aggregate function to the expression.

Example

This is best illustrated with an example.

You want to know how many employees in each department make between \$30,000 and \$40,000, how many make between \$40,000 and \$50,000, how many make between \$50,000 and \$60,000, and so on. To do this, you want a crosstab where each row corresponds to a \$10,000 range of salary.

The first step is to determine the expression that, given a salary, returns the next smaller salary that is a multiple of \$10,000. For example, given a salary of \$34,000, the expression would return \$30,000, and given a salary of \$47,000, the expression would return \$40,000. You can use the Int function to accomplish this, as follows:

```
int(salary/10000) * 10000
```

That expression divides the salary by 10,000 and takes the integer portion, then multiplies the result by 10,000. So for \$34,000, the expression returns \$30,000, as follows:

```
34000/10000 = 3.4
int(3.4) = 3
3 * 10000 = 30000
```

With that information you can build the crosstab. The following uses the Employee table in the Powersoft Demo Database:

- 1 Build a crosstab and retrieve the Dept\_ID and Salary columns.
- 2 In the Crosstab Definition dialog box, drag the Dept\_ID column to the Columns box.
- 3 Drag the Salary column to the Rows box and to the Values box and edit the expressions.

In the Rows box, use:

```
int(salary/10000) * 10000
```

In the Values box, use:

```
count(int(salary/10000) * 10000 for crosstab)
```

**FOR INFO** For more on providing expressions in a crosstab, see "Using expressions" on page 439.

- 4 Click OK.

This is the resulting workspace:

<b>Number of employees by department and salary</b> \$30,000 includes up to \$39,999		<b>Total number of employees making the salary</b>	
<b>Header [1] ↑</b>		<b>Department id</b>	
<b>Salary display@</b>			
<b>Header [2] ↑</b>		<b>row_column</b>	
		val crosstabsum( 1 )	
<b>Detail ↑</b>			
<b>Total number of employees in the department</b>		sum(val for	
<b>Summary ↑</b>			
<b>Footer ↑</b>			

And here is the crosstab during execution:

<b>Number of employees by department and salary</b> \$30,000 includes up to \$39,999							<b>Total number of employees making the salary</b>
		<b>Department id</b>					
<b>Salary</b>	<b>100</b>	<b>200</b>	<b>300</b>	<b>400</b>	<b>500</b>		
<b>\$20,000</b>				2	5	7	
<b>\$30,000</b>	3	8	2	5	2	20	
<b>\$40,000</b>	6	5	2	5	1	19	
<b>\$50,000</b>	4	3	3	2	1	13	
<b>\$60,000</b>	4	1		2		7	
<b>\$70,000</b>	2	1	1			4	
<b>\$80,000</b>	2	1				3	
<b>\$90,000</b>	1					1	
<b>\$130,000</b>			1			1	
<b>Total number of employees in the department</b>	22	19	9	16	9		

You can see, for example, that two people in department 400 and five in department 500 make between \$20,000 and \$30,000.



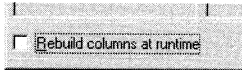
## Creating static crosstabs

By default, crosstabs are dynamic: when you run them, InfoMaker retrieves the data and dynamically builds the columns and rows based on the retrieved data. This is usually what you want. For example, if you have defined a crosstab that computes sales of printers and a new printer type is entered in the database after you defined the crosstab, you want the new printer to be in the crosstab. That is, you want InfoMaker to dynamically build the rows and columns based on current data, not the data that existed when the crosstab was defined.

Occasionally, however, you might want a crosstab to be static. That is, you want its columns to be established when you define the crosstab. You do not want additional columns to display in the crosstab during execution; no matter what the data looks like, you don't want the number of columns to change. You only want the updated statistics for the predefined columns. Here is how to do that.

❖ **To create a static crosstab:**

- 1 In the Crosstab Definition dialog box, clear the Rebuild Columns At Runtime checkbox:



- 2 Define the data for the crosstab as usual.
- 3 Click OK.

### What happens

With the checkbox cleared, instead of immediately building the crosstab's structure, InfoMaker first retrieves the data from the database. Using the retrieved data, InfoMaker then builds the crosstab structure and displays the workspace. It places all the values for the column specified in the Columns box in the workspace. These values become part of the crosstab's definition.

For example, in the following screen, the four values for Quarter (Q1, Q2, Q3, and Q4) are displayed in the workspace:

Sum Of Units	Quarter				
<b>Header [1] ↑</b>					
Product	Q1	Q2	Q3	Q4	Grand Total
<b>Header [2] ↑</b>					
product	units	units_1	units_2	units_3	crosstabsum(1)
<b>Detail ↑</b>					
"Grand Total"	sum(units	sum(units	sum(units	sum(units	sum(grand_sum_units for
<b>Summary ↑</b>					
<b>Footer ↑</b>					

At execution time, no matter what values are in the database for the column, the crosstab will show only the values that were specified when the crosstab was defined. In the printer example, the crosstab will always have the four columns it had when it was first defined and previewed in the Report painter.

**Making changes**

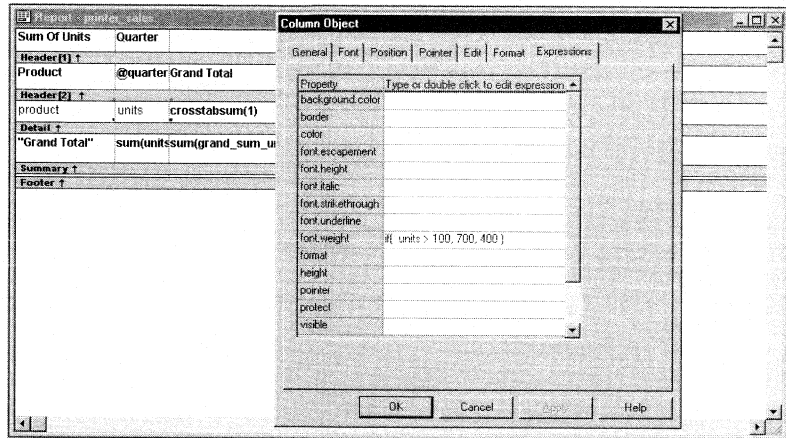
You can modify the properties of any of the columns in a static crosstab. You can modify the properties of each column individually, since each column is displayed in the workspace as part of the crosstab's definition. For example, in the printer crosstab you can directly modify the way values are presented in each individual quarter, since each quarter is represented in the workspace (the values are shown as units, units\_1, units\_2, and units\_3).

**Using property conditional expressions**

As with other reports, you can specify property conditional expressions to modify properties at execution time. You can use them with either dynamic or static crosstabs. With dynamic crosstabs, you specify an expression once for a column or value, and InfoMaker assigns the appropriate properties when it builds the individual columns during execution. With static crosstabs, you have to specify an expression for each individual column or value because the columns are already specified at definition time.

## Example

In the following crosstab, an expression has been specified for Units:



Here is the expression, which is for the Font.Weight attribute:

```
if (units > 100, 700, 400)
```

The expression specifies to use bold font (weight = 700) if Units is greater than 100. Otherwise, use normal font (weight = 400).

Here is the crosstab during execution:

Sum Of Units	Quarter				
Product	Q1	Q2	Q3	Q4	Grand Total
Cosmic	64	<b>104</b>	<b>134</b>	<b>160</b>	<b>462</b>
Galactic	9	21	13	12	<b>55</b>
Stellar	45	51	60	90	<b>246</b>
<b>Grand Total</b>	<b>118</b>	<b>176</b>	<b>207</b>	<b>262</b>	<b>763</b>

The large values are shown in bold.

FOR INFO For more about property conditional expressions, see Chapter 8, "Highlighting Information in Reports and Forms".



About this chapter

This chapter explains how to create reports using the RichText presentation style.

Contents

<b>Topic</b>	<b>Page</b>
What is rich text?	458
Using the RichText presentation style	459
Formatting keys and toolbars	471

## What is rich text?

Rich text format (RTF) is a standard for specifying formatting instructions and document content in a single ASCII document. An editor that supports rich text format interprets the formatting instructions and displays the text with formatting. If you look at rich text in a plain ASCII editor, you see complex instructions that are not very readable. The actual text of the document is buried amid all the formatting:

```
{\par}\pard\ql{\f2\fs18\cf0\up0\dn0 A RichText piece of text }
```

The same sample displayed without the commands looks like this:

A RichText piece of text

### Elements of rich text

Rich text in InfoMaker can have:

- ◆ Margins and tab settings for each paragraph
- ◆ Character formatting such as italic, bold, underline, or superscripts for each character
- ◆ Named input fields associated with database columns or other data
- ◆ Bitmaps
- ◆ A header and footer for the document

You can use toolbars, editing keys, and a popup menu to specify formatting. A print preview lets you view a reduced image of the document to see how it fits on the page.

### What is not supported

InfoMaker supports version 1.2 of the RTF standard, except for the following:

- ◆ No formatted tables
- ◆ No drawing objects
- ◆ No double-underline

## Using the RichText presentation style

The RichText presentation style allows you to combine input fields that represent database columns with formatted text. This presentation style is useful for display-only reports, especially mail-merge documents.

In the design workspace, you see the text along with placeholders called input fields:

```
{FNAME} {LNAME}
{COMPANY_NAME}
{ADDRESS}
{CITY}, {STATE} {ZIP}
```

```
Dear {FNAME}:
```

```
. . .
```

When you preview and print the report, InfoMaker replaces the input fields with values from the database. In this sample, you see the same text as you saw in the workspace, but now the input fields are replaced with values from the database:

```
Beth Reiser
AMF Corp.
1033 Whippany Road
New York, NY 10154
```

```
Dear Beth:
```

```
. . .
```

### Document template

The formatted text acts like a document template. There is only one copy of the text. As the user scrolls from row to row, the data for the current row is inserted in the input fields and the user sees the document with the current data. If the user edits the text, the changes are seen with every row of data.

### Input fields

In the RichText presentation style, an input field is associated with a column or computed field. It gets its value from the retrieved data or from the computed field's expression.

If an input field is not a computed field and its name does not match a column, there is no way to specify data for the input field.

There can be more than one copy of an input field in the rich text. In the sample above, there are two instances of the field FNAME. Each instance of the field displays the same data.

Unavailable settings

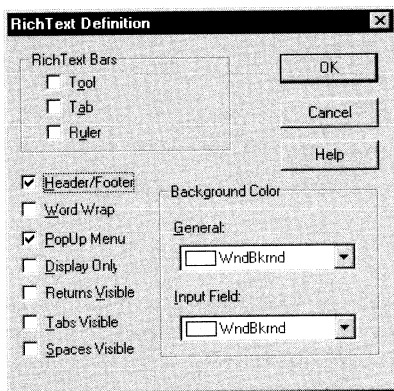
Not all the settings available in other report styles are available. You cannot apply code tables and edit styles, such as a DropDownDataWindow or EditMask, to input fields. You cannot use slide left and slide up settings to reposition input fields automatically.

## Creating the report

❖ **To create a RichText report:**

- 1 In the Report painter, create a new report and select the RichText presentation style.
- 2 Select data for the report as you do for any report.  
The columns become input fields.
- 3 Leave the SQL Select painter and return to the Report painter workspace.

InfoMaker displays the RichText Definition dialog box:



- 4 Specify settings for the report in the RichText Definition dialog box, and click OK.

Available settings

This table describes the types of settings you can make for the RichText report:

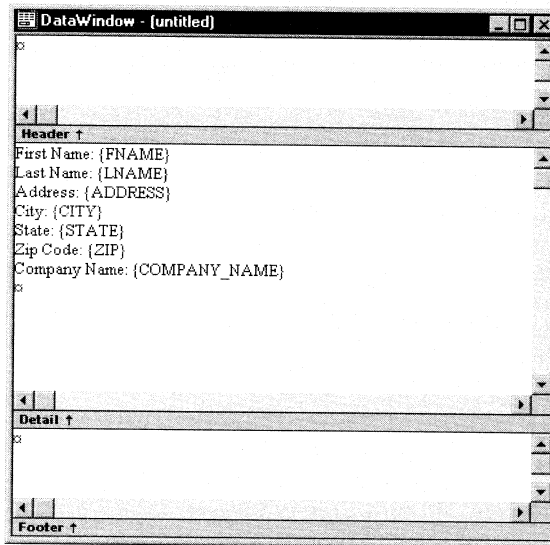
You can specify	With these settings
Tools available	Rich text bars: Tool, Tab, and Ruler, and PopUp Menu
Whether various nonprinting characters are visible	Returns Visible, Tabs Visible, and Spaces Visible



You can specify	With these settings
Colors for the whole background and the background of input fields	Background Color: General and Input Field
Whether you are prevented from editing input fields and text	Display Only
Whether newly entered text will wrap within the display	Word Wrap
Whether there will be a header and footer for the printed report	Header/Footer

### Editing the content

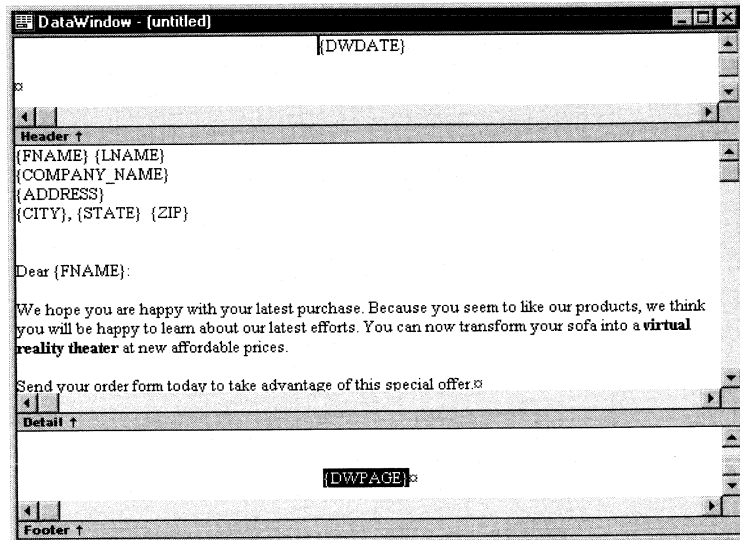
After you click OK in the RichText Definition dialog box, you see input fields with their labels in the detail band of the report:



You can:

- ◆ Begin editing text in the detail, header, or footer bands, building a report around the input fields. You can delete, move, copy, and paste text and input fields as needed.
- ◆ Include a rich text file you have already prepared. If you include a rich text file created in PowerBuilder that contains input fields, those names should match the columns selected in the report.
- ◆ Add computed fields that will appear as input fields in the report and whose value comes from the computed field expression.

This sample shows how you might rearrange the input fields in a sales letter:



## Editing text

You can add text by typing directly in the workspace. You don't have to create text objects as you do for other report styles. The Report painter's StyleBar lets you apply formatting to selected text. The RichText toolbars are *not* available in the painter.

---

### Preview mode and editing text

It may seem convenient to edit text in preview mode, because the toolbars are available. However, *any changes you make to the text when previewing are temporary*. They are discarded as soon as you return to the workspace.

---

## Inserting a file

If you have a rich text file, you can include it in the report. You can insert text from a file into the detail, header, or footer band.

### ❖ To insert a file:

- 1 Click in the text in any band to set the insertion point for the file.

- 2 Right-click in the workspace and select Insert File from the popup menu.
- 3 Select the file you want to insert in the file selection dialog box.

Only the body of the file is used. If the file has a header or footer, it is ignored.

## Headers and footers

You decide whether your RichText report has a header and footer by checking Header/Footer in the RichText Definition dialog box or Rich Text Object property sheet (described in "Formatting for RichText objects within the report" next).

To display a page number or a date in the header or footer, you can insert the predefined computed fields **Page n of n** or **Today()**.

---

### Losing header and footer text

If you clear the Header/Footer checkbox, any text in the header and footer is discarded.

---

## Formatting for RichText objects within the report

Each type of object in a RichText report has its own property sheet. When you select Properties from the popup menu, the property sheet you get depends on what is selected.

Most of the objects in a RichText report correspond to familiar objects, like bitmaps, columns, and computed fields. You can also specify formatting for a temporary *selected text object*. In a RichText report, the objects are:

- ◆ The whole document
- ◆ Selected text and paragraphs
- ◆ Input fields (associated with columns or computed fields)
- ◆ Pictures

This section describes how to select each type of object and access its property sheet. The user can access the property sheets too if you enable the Popup Menu option on the Rich Text Object's General property sheet.

## The whole RichText object

Settings for the whole RichText report include:

- ◆ Tools
- ◆ Margins for printing
- ◆ Whether the document has a header and footer
- ◆ Display of nonprinting characters
- ◆ Whether pictures are displayed or represented by empty frames
- ◆ Standard report settings such as units of measurement and the pointer

You specified values for some of these in the initial RichText Definition dialog box. Use the following procedure to change settings.

### ❖ To make general settings for the report:

- 1 Make sure nothing is selected in the workspace by clicking to set the insertion point.
- 2 Right-click in the workspace and select Properties from the popup menu.  
The RichText Object property sheet displays.
- 3 Click Help to get more information about a specific setting.

## Selected text and paragraphs

You can specify detailed font formatting for selected text. The selected text can be one character or many paragraphs.

If an input field is part of the selection, the font settings apply to it too. A picture that is part of the selection ignores settings for the selected text object.

### ❖ To specify formatting for selected text:

- 1 Select the text you want to format.
- 2 Right-click in the workspace and select Properties from the popup menu.

The Selected Text Object property sheet displays. You can set:

- ◆ **Paragraph alignment** The alignment setting on the Selected Text page applies to all paragraphs in the selection.
- ◆ **Font formatting** Settings on the Font page apply to text in the selection, including input fields.

Paragraphs	There are also settings for selected paragraphs. You can display the Paragraph dialog box by pressing CTRL+SHIFT+S. The user can double-click the ruler bar or press the key combination to display the same dialog box.
Default font	You can change the default font by double-clicking on the toolbar or pressing CTRL+SHIFT+D. You can't change the default font in the painter.

## Input fields

An input field can be either a column or computed field. Before you retrieve data, its value is shown as two question marks (??).

The text can include many copies of a named input field. The same data will appear in each instance of the input field.

Column input fields	The columns you select for the report become input fields in the rich text. Because the input field's name matches the column name, InfoMaker displays the column's data in the input field.
---------------------	--

If an input field exists in the text, you can copy and paste it to create another copy. If you need to recreate a column input field that you deleted, use this procedure.

❖ **To insert a column input field in the text:**

- 1 Select Objects>Column from the menu bar.
- 2 Click in the text where you want the column input field to appear.  
InfoMaker displays a list of the columns selected for the report.
- 3 Select a column for the input field.

Properties for input fields	An input field is selected when the whole field is flashing. If there is a solid selection with a flashing insertion point, then the current object is selected text and you'll get a different property sheet.
-----------------------------	---

---

### Editing input field contents

When an input field is flashing, press the space bar to activate the input field and edit the data. In preview mode, you can edit the input field, but you cannot update the database with your changes.

---

❖ **To set properties for an input field:**

- 1 Click on the input field so it is flashing.
- 2 Display the popup menu and select Properties.

- 3 On the Font page, specify text formatting.
- 4 On the Format page, specify a display format.

If there are multiple copies of an input field, the format settings apply to all the copies. Background color on the Font page applies to all input fields. Other settings on the Font page apply to individual instances.

**Computed field input fields** When you display the property sheet for a computed field, the settings are a little different. You can specify the input field name and its expression on the Compute page.

#### Computed fields

Computed fields have an expression that specifies the value of the computed field. In rich text, they are represented as input fields too. You specify a name and an expression. The data value comes from evaluating the expression and cannot be edited.

#### ❖ To define a computed field:

- 1 From the Objects menu, select Computed Field or select one of the predefined computed fields at the bottom of the Objects menu.

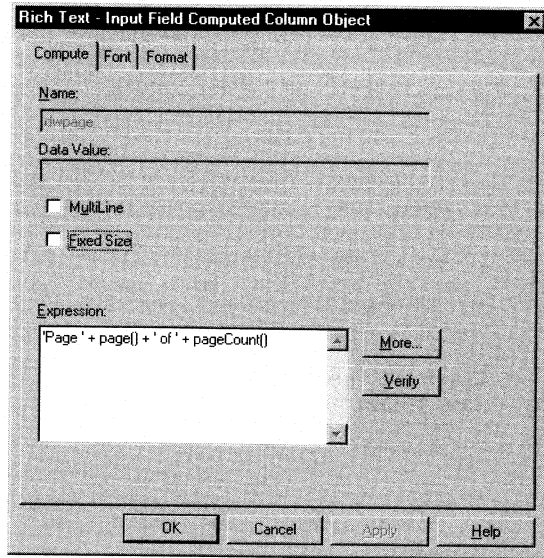
---

#### **Predefined computed fields**

InfoMaker provides several predefined computed fields, but in a RichText report only the page number (Page n of n) and today's date (Today()) are available.

---

- 2 Click in the text where you want the computed field to appear.  
InfoMaker displays the property sheet for the computed field:



- 3 On the Compute page, name the computed field and specify its expression.
- 4 (Optional) On the Font page, specify text formatting.
- 5 (Optional) On the Format page, specify a display format.

If there are multiple copies of a computed field input field, the expression and format settings apply to all the copies. Font settings apply to individual instances.

For more about computed field expressions and display formats, see Chapter 5, "Enhancing Reports".

## Pictures

You can include bitmaps (BMP files) in rich text.

### ❖ To insert a picture in the rich text:

- 1 Select Objects>Picture from the menu bar.
- 2 Click in the text where you want the picture to appear.  
InfoMaker displays the Select Picture dialog box.

- 3 Select the file containing the picture.

A picture is selected when you can see its nine sizing handles. (If the picture is very small, the handles converge and the selected picture looks black.) When it is part of a text selection, it displays with inverted colors.

❖ **To specify settings for the picture:**

- 1 Click on the picture so you see its resizing handles.
- 2 Right-click in the workspace and select Properties from the popup menu.

The Bitmap Object property sheet displays.

You can change the size and position of the picture by dragging its handles. You can also change the size using the Bitmap Object dialog box:

<b>Choosing this Picture Size Option</b>	<b>Has this effect</b>
Original size	The bitmap displays at its original size. You can set the x and y offset to make additional space above and to the left of the picture
Resizable	The bitmap fills the picture dimensions
Resizable - Maintain X/Y Ratio	The bitmap fills the picture dimensions as well as it can while maintaining the aspect ratio of the picture

❖ **To change the picture's dimensions:**

- ◆ Drag any handle on the edge of the selected picture.

*or*

Change the numbers in the Picture Dimension edit boxes on the Bitmap property page.

The picture will fill the frame according to the Picture Size Option you've selected.

❖ **To change the picture's position relative to the text baseline:**

- ◆ Select the picture and drag the handle in the center of the picture up or down.

You can drag the picture above or below the text baseline. You can't drag the picture outside the paragraph. To do that, cut and paste the picture somewhere else in the text.

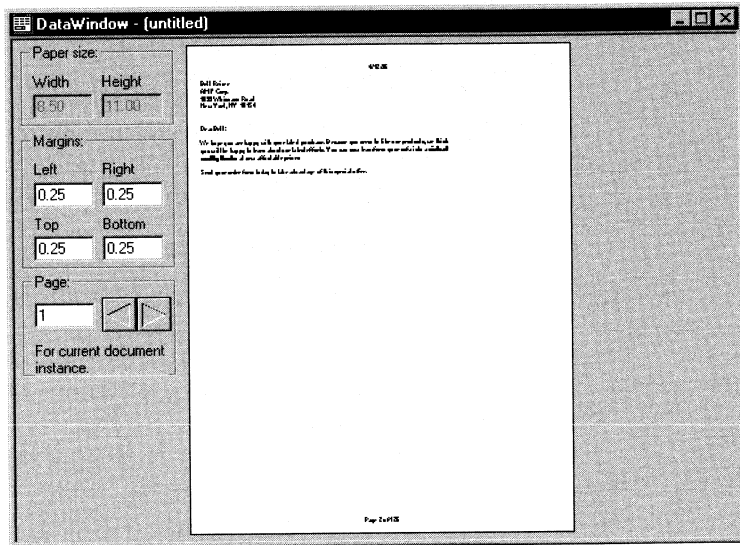


## Previewing and printing

- Report preview      You can preview the RichText report so you can see what it looks like with data.
- ❖ **To preview the report:**
    - ◆ Click the Preview button on the toolbar.  
*or*  
Select Preview>Design from the menu bar.
  - ❖ **To view the header and footer text:**
    - 1 From the Design menu, select Header/Footer.
    - 2 Select Header/Footer again to go back to the detail band.
- Changes in preview mode      **Data** While you are previewing the report, you can use the scroll buttons in the Preview toolbar to move from row to row and you can change data in the input fields. The changes you make, however, do not affect the data in the database.
- Text** Any changes you make to the rich text *will be discarded* when you return to the workspace. Any changes that you want to keep must be made in the workspace, not in preview.
- If the Display Only setting is checked, you cannot change text or data in preview mode.
- Print Preview      Print Preview displays a reduced view of one row of data as it would be printed.

❖ **To see the report in Print Preview:**

- ◆ While in preview mode, select File>Print Preview from the menu bar:



In Print Preview, you can test different margin settings and scroll through the pages of the document.

You *cannot* scroll to view other rows of data.

Any changes you make to settings in Print Preview are discarded when you return to the workspace.

---

### Setting margins

To specify permanent margin settings for the RichText report, use the Print Specifications page of the Rich Text Object property sheet.

---

## Formatting keys and toolbars

When the toolbar is visible, you can use its buttons to format text. The changes you make in preview are temporary.

The keystrokes listed in the following tables also assign formatting to selected text. You can use these in the workspace and save the changes permanently.

### On Macintosh

On the Macintosh, use the **COMMAND** key instead of the **CTRL** key.

Using the clipboard	Key
Cut	CTRL+X, DELETE (when there is a selection)
Paste	CTRL+V, SHIFT+INSERT
Copy	CTRL+C
Undo	CTRL+Z

Assigning font attributes	Key
Bold	CTRL+B
Italic	CTRL+I
Underline	CTRL+U
Subscript	CTRL+=
Superscript	CTRL+SHIFT+=
Strikeout	CTRL+K
Change font	CTRL+SHIFT+D or double-click on empty part of toolbar

Setting line spacing	Key
Single space	CTRL+1
Double space	CTRL+2
One and a half space	CTRL+5

Aligning text	Key
Justify	CTRL+J

<b>Aligning text</b>	<b>Key</b>
Center	CTRL+E
Left	CTRL+L
Right	CTRL+R
Set paragraph formatting	CTRL+SHIFT+S or double-click the ruler

<b>Editing</b>	<b>Key</b>
Insert a new paragraph	ENTER
Insert an empty line	CTRL+N
Delete character to right of insertion point	DELETE
Delete character to left of insertion point	BACKSPACE

<b>Input fields</b>	<b>Key</b>
Select the input field at the insertion point	ENTER
Activate the input field at the insertion point	SPACE
When input field is active, accept data and exit field	ENTER
When input field is active, exit field without changing data	ESC
Move to next input field	CTRL+TAB
Move to previous input field	SHIFT+CTRL+TAB

<b>Miscellaneous</b>	<b>Key</b>
Select All	CTRL+A
Print	CTRL+P
Undo	CTRL+Z
Toggle display of nonprinting characters	CTRL+*
Toggle preview mode	CTRL+F2

<b>Navigating and selecting text</b>	<b>Navigating key</b>	<b>Selection key</b>
Move or select a character to the right or left	RIGHT ARROW OR LEFT ARROW	SHIFT+RIGHT ARROW OR SHIFT+LEFT ARROW
Move or select a word to the right or left	CTRL+RIGHT ARROW OR CTRL+LEFT ARROW	CTRL+SHIFT+RIGHT ARROW OR CTRL+SHIFT+LEFT ARROW
Move or select a line up or down	UP ARROW OR DOWN ARROW	SHIFT+UP ARROW OR SHIFT+DOWN ARROW
Move or select to start of line	HOME	SHIFT+HOME
Move or select to end of line	END	SHIFT+END
Move or select to start of document	CTRL+HOME	CTRL+SHIFT+HOME
Move or select to end of document	CTRL+END	CTRL+SHIFT+END
Move to next input field	CTRL+TAB	
Move to previous input field	SHIFT+CTRL+TAB	



# Using OLE in a Report

## About this chapter

This chapter discusses OLE objects in reports. You can select OLE as the presentation style when you build a new report, or you can mix OLE objects with columns and other reports. The procedures in this chapter explain how to use OLE objects and specify their behavior.

---

**Platform note**

OLE is supported on the Windows platform only.

---

## Contents

<b>Topic</b>	<b>Page</b>
OLE support in reports	476
OLE objects and OLE presentation style	478
Using OLE columns in a report	489

## OLE support in reports

A report can include an object that is a container for an OLE object. The container stores information about the application that created the object and it can launch the application to display or modify the OLE object.

The container can fill the whole report, when you create a new report using the OLE presentation style, or it can exist alongside other objects in a report, when you add an OLE object to an existing report. You can also read OLE data from a blob column in a database and display the objects in the report. These mechanisms support both OLE 2.0 and OLE 1.0 servers.

You can use OLE objects in reports in the following ways:

- ◆ **OLE object in a report** The OLE object is displayed in its container object with the report data and other objects, such as bitmaps or text. You can associate it with data in a particular row, the rows on a page, or with all rows. You choose which columns in the report are transferred to the OLE object. You can add an OLE container object to a report that uses any presentation style that supports multiple reports (this does not include the graph and RichTextEdit presentation styles).
- ◆ **OLE presentation style** The OLE presentation style is similar to an OLE object in a report. The difference is that the OLE container is *the only* object in the report. The underlying data is *not* presented in column objects and *there are no other* objects, such as bitmaps or text. The OLE object is *always* associated with all the rows in the report.
- ◆ **OLE database blob column** OLE objects that are stored in the database in a blob column are displayed in each row of the report.

You can also add OLE custom controls (OCXs) to reports. OCXs range from simple visual displays, such as meters and clocks, to more complex controls that perform spellchecking or image processing.

## About activation

When you're working in the Report painter, you can start the server application for an OLE object by double-clicking the object. The OLE object is activated in place if the server supports in-place activation. Once the server application has started, you can use the tools provided by the server to edit the initial presentation of the object.



**In-place or offsite activation**

When an OLE object is activated in place, the menus and toolbars of the server application merge with or replace those of the application from which it was launched. If the server application does not support in-place activation, it is activated offsite, which means that the server application opens and the object becomes an open document in the server's window.

---

In InfoMaker, when you preview a report that has one or more OLE objects associated with it, you cannot activate the OLE objects. When you preview a report, InfoMaker retrieves data from the database and displays the report as it will appear when printed. For this reason, activation is not possible.

To activate an OLE object, you first need to add the report that contains the object to a form. Once you've done this, you can double-click the OLE object to activate the server application.

If the OLE object is associated with *all rows* retrieved and is in the foreground or background layer, not the band layer, you can activate the object. If the object is associated with a single row or page or is in the band layer, you can see the object but can't activate it. Reports created using the OLE presentation style are *always* associated with all rows.

Unlike OLE objects, OCXs are always active. They do not contain objects that need to be opened or activated by double-clicking.

**What's next**

Whether you are inserting an OLE object into an existing report or using the OLE presentation style, you use the same procedures to define, preview, and specify data for the OLE object. Because of their similarities, the next section discusses both OLE objects in existing reports and the OLE presentation style. The last section discusses OLE database blob columns.

## OLE objects and OLE presentation style

The OLE object in a report and the OLE presentation style are very similar. Both are OLE container objects within the report. They have these characteristics in common:

- ◆ **Icon or contents** The report can display the OLE object as an icon or it can display an image of the contents when display of contents is supported by the server.
- ◆ **Data from the report** You specify which report columns you want to transfer to the OLE object. The data that is sent to the OLE server replaces the OLE object template specified in the painter.

The OLE object in a report and the OLE presentation style have these main differences:

- ◆ **Associating the object with rows** When the OLE object is added in the Report painter, it can be associated with individual rows, groups of rows, or all rows. In the presentation style, the OLE object is associated with all rows.
- ◆ **Property sheet** The property sheet for the OLE object has no settings for the report because the report has its own property sheet. The property sheet for the OLE presentation style includes settings for the report and the OLE container object.

---

### Not all servers are appropriate

The features of the OLE server application determine whether it can provide useful information in a report.

If the server doesn't support display of contents, it won't be useful for objects associated with rows. The user will only see the icon. Some servers support the display of contents, but the view is scaled too small to be readable even when the object is activated.

---

In this section

This section includes procedures for:

- ◆ Adding an OLE object to a report
- ◆ Using the OLE presentation style
- ◆ Defining the OLE object
- ◆ Previewing the report
- ◆ Specifying report data for the OLE object

## Adding an OLE object to the report

The procedures for adding OLE objects and OLE custom controls to a report are similar. Both exist within the report with other objects, such as columns, computed fields, and text objects.

You use the first procedure in this section whether you want to add an OLE object or an OCX to an existing report. The procedure opens the Insert Object dialog box in which you define the OLE object.

### ❖ To place an OLE object in a report:

- 1 Open the Report painter and select the report that will contain the OLE object.

The Report painter workspace displays the report.

- 2 From the toolbar, click the Object dropdown arrow and select the OLE button (not OLE Database Blob).

*or*

Select Objects>OLE Object from the menu bar.

- 3 Click where you want the OLE object.

InfoMaker displays the Insert Object dialog box.

**FOR INFO** To use the Insert Object dialog box, see "Defining the OLE object" on page 480 .

## Using the OLE presentation style

Use the OLE presentation style to create a report that consists of a single OLE object. The following procedure creates the new report and opens the Insert Object dialog box.

### ❖ To create a new report using the OLE presentation style:

- 1 Open the Report painter and select New in the Select Report dialog box.

The New Report dialog box displays.

- 2 Select a data source and the OLE presentation style, then click OK.

You are prompted to specify the data.

- 3 Specify the data you want retrieved into the report.

**FOR INFO** For more information about selecting data, see Chapter 4, "Defining Reports".

- 4 Click OK.

InfoMaker displays the Insert Object dialog box in which you define the OLE object.

**FOR INFO** To use the Insert Object dialog box, see "Defining the OLE object" next.

## Defining the OLE object

You define the OLE object in the Insert Object dialog box. It has three tab pages:

<b>If you want to</b>	<b>Select</b>
Embed an OLE server object in the report	Create New
Link or embed the contents of an existing file as an OLE object so that it can be activated using the application that created it	Create From File
Insert an OCX in the report	Insert Control

This section contains procedures for each of these selections.

### Create New

Use the following procedure if you want to embed a new OLE server object.

#### ❖ To embed a new OLE server object using the Create New tab:

- 1 Select the Create New tab.
- 2 In the Object Type box, highlight the OLE server you want to use.  
You can click Browse to get information about the server from the registry.
- 3 Optionally display the OLE object as an icon by doing one of the following:
  - ◆ Check Display as Icon to display the server application's default icon in the control.
  - ◆ Check Display as Icon and then select Change Icon to supply a nondefault icon and icon label.

- 4 Click OK.

The OLE object is inserted in your document and the OLE server is activated (in place if the server supports in-place activation). You can edit the object now. However, this object only provides the initial presentation of the OLE object. The presentation is replaced by data from the report when you preview the report.

- 5 Click outside the hatched border of the OLE object to deactivate it.

The Report painter menus replace the OLE server's menus and InfoMaker displays the property sheet for the OLE object with the Data property page selected.

- 6 On the Data property page, specify how the OLE object will use the report's data.

You can drag the columns you want the OLE object to use to the Target Data box. You can also edit columns and control the grouping of data. If you want to, you can set or change these specifications later.

**FOR INFO** For more information, see "Specifying data for the OLE object" on page 484.

- 7 If you are inserting an OLE object in an existing report and you want to associate the object with the current row, click the Position tab and change the value in the Layer box to Band.

You cannot change this value to Band if you are using the OLE presentation style.

- 8 Click OK when you have finished.

The OLE object is either empty or contains the initial presentation.

#### Create From File

Use the following procedure if you want to link or embed the contents of an existing file as an OLE object so that it can be activated using the application that created it. Most of the steps in this procedure are the same as those for embedding a new OLE server object.

#### ❖ To link or embed an existing object using the Create From File tab:

- 1 Select the Create From File tab.
- 2 Specify the filename in the File Name box. If you do not know the name of the file, click the Browse button and select a file in the dialog box.
- 3 To create a link to the file, rather than embed a copy of the object in the control, select the Link checkbox.
- 4 Go to step 3 in the procedure for the Create New tab (above) to continue.

## Insert Control

Use the following procedure if you want to insert an OLE custom control (OCX) in the report.

### ❖ To insert an OCX using the Insert Control tab:

- 1 In the Control Type box, highlight the OCX you want to use.

To get information about the selected OCX, you can click **Browse**. OCXs are self-documenting. InfoMaker gets the property, event, and function information from the OCX itself via the registry.

- 2 If the OCX you want has not been registered, click **Register New**.

You will be prompted for the file that contains the registration information for the OCX.

- 3 Click **OK**.

InfoMaker displays the property sheet for the control.

- 4 Change the control properties if desired and click **OK**.

InfoMaker displays the property sheet for the OLE object with the **Data** property page selected.

- 5 On the **Data** property page, specify how the OLE object will use the report's data.

If you have inserted an OCX that does not display data, such as the **Clock** control, you don't need to transfer data to it.

**FOR INFO** For more information, see "Specifying data for the OLE object" on page 484.

- 6 Click **OK** when you have finished.

## Previewing the report

Previewing the report lets you see how the OLE object displays the data from the report.

### ❖ To preview the report with the OLE object:

- ◆ Click the **Preview** toolbar button.

*or*

Select **Design>Preview** from the menu bar.

The report retrieves rows from the database and replaces the initial presentation of the OLE object with an image of the data that the OLE server provides.

**Tip**

If you have previewed already, you may need to click the Retrieve button to retrieve the rows again. This resets the data being transferred to the OLE object.

---

**Activating and editing the OLE object**

In the painter

InfoMaker stores an initial presentation of the OLE object that it displays before data is retrieved and in newly inserted rows. When you activate the OLE object in the painter, you are editing the initial presentation of the OLE object. Any changes you make and save affect only this initial presentation. After rows are retrieved and data transferred to the OLE object, an object built using the data replaces the initial presentation.

In preview

InfoMaker displays the initial presentation of the OLE object while it is retrieving rows and then replaces it with the retrieved data. You cannot activate the OLE object when you preview the report. If you add the report to a form, you can activate the OLE object when you run the form.

You can save the object with its data by saving the report as a Powersoft report (PSR).

FOR INFO For more information, see "About activation" on page 476.

**❖ To activate the OLE object in the container in the painter:**

- ◆ Double-click the object.

*or*

Select Open from the container's popup menu.

When you double-click an OCX, it displays its property sheet (which is different from the report property sheet for the container). OCXs are always active.

**Changing the object in the control**

In the Report painter, you can change or remove the OLE object in the OLE container object.

**❖ To delete the OLE object in the container:**

- ◆ Select Delete from the container's popup menu.

The container object is now empty and cannot be activated.

❖ **To change the OLE object in the container:**

- 1 Select Insert from the container's popup menu.  
InfoMaker displays the Insert Object dialog box.
- 2 Choose one of the tabs and specify the type of object you want to insert, as you did when you defined the object.
- 3 Click OK.

## Specifying data for the OLE object

You specify data for the OLE object on the Data property page in the property sheet for the container.

When you insert a new object, just after you deactivate the object, InfoMaker displays the property sheet with the Data property page selected. If you want to change the settings later, use the following procedure.

❖ **To change the data settings:**

- ◆ Select Properties from the container's popup menu and select the Data tab.

What the data is for

When an OLE object is part of a report, you can specify that some or all of the data the report retrieves be transferred to the OLE object too. You can specify expressions instead of the actual columns so that the data is grouped, aggregated, or processed in some way before being transferred.

The way the OLE object uses the data depends on the server. For example, data transferred to Microsoft Excel is displayed as a spreadsheet. Data transferred to Microsoft Graph populates its datasheet, which becomes the data being graphed. Some OCXs don't display data, so you would not transfer any data to them.

## Data property page

Three boxes on the Data property page list data columns or expressions. This table explains what they are for:

<b>Data property page item</b>	<b>Contents and purpose</b>
Source Data	Lists the columns retrieved in the report You can't change the source data



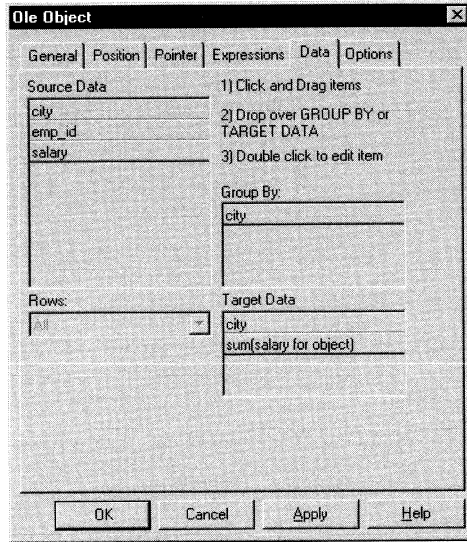
Data property page item	Contents and purpose
Group By	<p>Specifies how InfoMaker groups the data it transfers to the OLE object</p> <p>Aggregation functions in the target data expressions use the groupings specified here</p> <p>To populate the Group By box, drag columns to it from Source Data or Target Data. The same source column can appear in both Group By and Target Data</p>
Target Data	<p>Specifies the data that you want to transfer to the OLE object</p> <p>To populate the Target Data box, drag columns to it from Source Data or Group By</p> <p>If there is something in the Group By box, InfoMaker automatically converts each new column you drag to the Target Data box to an aggregate expression, unless the new column is also in the Group By box</p> <p>The order of the columns and expressions is important to the OLE server. You need to know how the server will use the data to choose the order</p> <p>An aggregation function uses the range <i>for object</i>. The expression will have a value for each group (or combination of groups) in the Group By box</p>

On the Data property page, you can also:

- ◆ Change the order of the grouping or the target data by dragging an item up or down within its box.
- ◆ Edit an item by double-clicking on it in the Target Data or Group By box. In the Modify Expression dialog box, you can edit the expression or use the Functions or Columns boxes and the operator buttons to select elements of the expression.

Example of a completed Data property page

This example of the Data property page specifies two columns to transfer to Microsoft Graph: city and salary. Graph expects the first column to be the categories and the second column to be the data values. The second column is an aggregate because the city column is in the Group By box:



Specifying a value for Rows

The last setting on the Data property page specifies how the OLE object is associated with rows in the report. The selection (all rows, current row, or page) usually corresponds with the band where you placed the OLE object, as explained in this table. If you used the OLE presentation style to create the report, you cannot change this setting: the OLE object is always associated with all the rows in the report.

Range of rows	When to use it
All	When the OLE object is in the summary, header, or footer band, or the foreground or background layer. For the OLE presentation style, All is the only choice  Rows must be All and Layer must be Foreground or Background if you want the user to be able to activate the object  Target data for all rows is transferred to the object

Range of rows	When to use it
Current Row	<p>When the OLE object is in the detail band</p> <p>There is an instance of the OLE object for every row. Target data for a single row is transferred to each object</p> <p>Because OCXs must be in the foreground or background layer, they cannot be associated with individual rows in the detail band</p>
Page	<p>When the OLE object is in the group header or trailer, foreground, or background</p> <p>Target data for the rows on the current page is transferred to the OLE object</p>

### Range of rows and activating the object

When the range of rows is Current Row or Page, the user *cannot* activate the OLE object. The user can see contents of the object in the form of an image presented by the server but cannot activate it.

If you want the user to activate the object, Rows must be set to All and Layer on the Position property page must be Foreground or Background.

## Additional settings on the property sheet

The Options property page for the OLE object's property sheet has some additional settings. This table describes the settings you can make:

Options property page setting	Effect
Client Name	<p>A name for the OLE object that some server applications use in the title bar of their window</p> <p>Corresponds to the ClientName Report property</p>
Contents	Whether the object in the OLE container is linked or embedded. The default is Any, which allows either method
Activation	<p>How the OLE object is activated:</p> <ul style="list-style-type: none"> <li>◆ <b>Double click</b> When the user double-clicks on the object, the server application is activated</li> </ul> <p>Activation is only possible when the report that contains the OLE object is embedded in a form</p>

<b>Options property page setting</b>	<b>Effect</b>
Display Type	What the OLE container displays. You can choose: <ul style="list-style-type: none"><li>◆ <b>Manual</b> Display a representation of the object, reduced to fit within the container</li><li>◆ <b>Icon</b> Display the icon associated with the data. This is usually an icon provided by the server application</li></ul>
Link Update	When the object in the OLE container is linked, the method for updating link information. Choices are: <ul style="list-style-type: none"><li>◆ <b>Automatic</b> If the link is broken and InfoMaker cannot find the linked file, it displays a dialog box in which the user can specify the file</li><li>◆ <b>Manual</b> If the link is broken, the object cannot be activated</li></ul>

## Using OLE columns in a report

You can create OLE columns in a report. An OLE column allows you to retrieve blob (binary large-object) data from a database into a report.

---

### Database support for OLE columns

If your database supports a blob data type, then you can implement OLE columns in a report. The name of the data type that supports blob data varies.

**FOR INFO** For information on which data types your DBMS supports, see your DBMS documentation.

---

## Creating an OLE column

This section describes how to create an OLE column in a report. The steps are illustrated using a table named `ole` in the Powersoft Demo Database. It contains three columns: `id`, `object`, and `description`:

- ◆ The `id` column is an integer and serves as the table's key
- ◆ The `object` column is a blob data type and contains OLE objects associated with several OLE servers
- ◆ The `description` column has a brief description of the object in each row

Creating the  
database table

For this sample procedure, you can use the `ole` table in the Powersoft Demo Database or you can create your own table. Follow this procedure to create your own table.

### ❖ To create the database table:

- 1 In the Database painter, create a table to hold the blob (binary large-object) data. The table must have at least two columns: a key column and a column with the blob data type.

The actual data type you choose depends on your DBMS. For example, in SQL Anywhere, choose long binary as the data type for the blob column. In SQL Server, choose Image.

**FOR INFO** For information about data types, see your DBMS documentation.

- 2 Define the blob columns as allowing NULLs (this allows you to store a row that doesn't contain a blob).

Adding a blob column to the report

This procedure describes how to add a blob column to a report.

❖ **To add a blob column to a new report:**

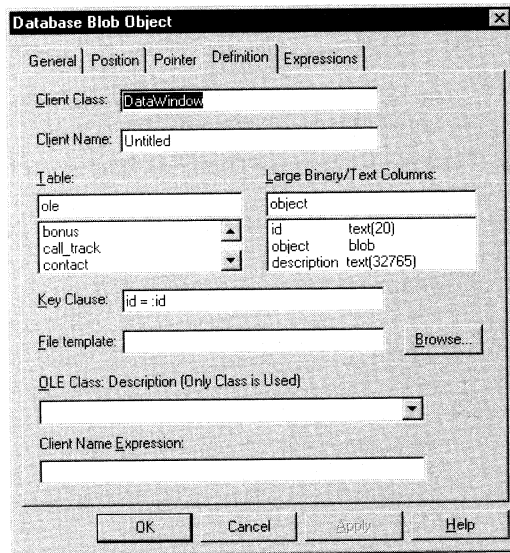
- 1 Open the Report painter and create a new report.
- 2 Specify the table containing the blob as the data source for the report.

Be sure to include the key column, but you cannot include the blob column in the data source. If you try, a message tells you that its data type is not supported. You will add the blob column later in the Report painter workspace. (If you use Quick Select, the blob column is not listed in the dialog box.)

In the example, you must choose the id column. Choosing the description column is optional.

- 3 Add the blob column to the report by selecting OLE Database Blob from the Objects menu and clicking the place in the report where you want the blob object.

The Blob Object property sheet displays with the Definition property page selected:



Setting properties for the blob column

This procedure describes the properties you need to set for the blob column.

❖ **To set properties for a blob column:**

- 1 (Optional) Enter the client class in the Client Class box. The default is DataWindow.

This value is used in some OLE server applications to build the title that displays at the top of the server window.

- 2 (Optional) Enter the client name in the Client Name box. The default is Untitled.

This value is used in some OLE server applications to build the title that displays in the title bar of the server window.

- 3 In the Table box, select the database table that contains the blob database column you want to place in the report.

The names of the columns in the selected table display in the Large Binary/Text Columns listbox.

- 4 In the Large Binary/Text Columns box, select the column that contains the blob data type from the list.

- 5 If necessary, change the default key clause in the Key Clause box.

InfoMaker uses the key clause to build the WHERE clause of the SELECT statement used to retrieve and update the blob column in the database. It can be any valid WHERE clause.

Use colon variables to specify report columns. For example, if you enter this key clause:

```
id = :id
```

the WHERE clause will be:

```
WHERE id = :id
```

- 6 Identify the OLE server application by doing one of the following:
  - ◆ If you always want to open the same file in the OLE server application, enter the name of the file in the File Template box.
 

For example, to specify a particular Word for Windows document, enter the name of the DOC file. If the file is not on the current path, enter the fully qualified name.

---

**Use the Browse button to find the file**

If you do not know the name of the file you want to use, click the Browse button to display a list of available files. Select the file you want from the resulting window.

---

- ◆ If you do not want to open the same file each time, select an OLE server application from the OLE Class: Description dropdown listbox.

If your OLE server application does not appear in this list, run the Windows RegEdit utility to add it.

**FOR INFO** For more information about RegEdit, see the Windows online Help REGEDIT.HLP and REGEDITV.HLP.

---

**When the server doesn't match the OLE blob data**

If you specify a server that doesn't match the OLE blob object or if your database contains objects belonging to different servers, the OLE mechanism can usually handle the situation. It looks for the server specified in the object and starts it instead of the server you specified.

---

- 7 Enter text or an expression that evaluates to a string in the Client Name Expression box.

The server may use this expression in the title of the window in the OLE server application. The expression you specify can identify the current row in the report.

---

**Use an expression to make sure the name is unique**

Use an expression to make sure the name is unique. For example, you might enter the following expression to identify a document (where ID is the integer key column):

```
"Document " + String(id)
```

---

- 8 (Optional) Select the General tab and enter a name for the object in the Name box. You don't need to name the object, but doing so allows you to refer to it in scripts.

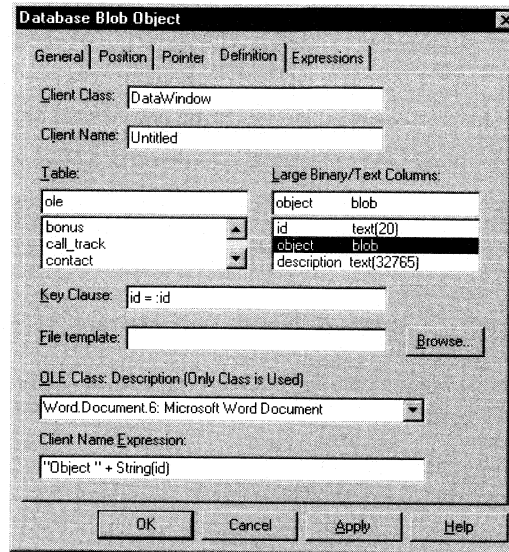
- 9 Click OK.

InfoMaker closes the Database Blob Object dialog box and displays the Report painter workspace. The blob column is represented by a box labeled Blob.

- 10 Save the report.



Here is the completed Database Blob Object dialog box for the example:



Making the blob column visible for OLE 1.0 servers

For OLE 1.0 servers, the blob column is invisible in the report until you activate the OLE server. To make it easy to find the blob column, you can place a drawing object behind the blob object. When the report displays, the drawing object will indicate the location of the blob column until the user double-clicks the column to open the server application.

The drawing object is not necessary for OLE 2.0 servers, which provide an icon for the object.

Previewing an OLE column

Before using the report in an application, you should preview it to see how it works.

❖ **To preview an OLE column:**

- ◆ Click the Preview button  
*or*  
Select Preview from the Design menu.

## Embedding reports in other applications

This section provides information about embedding InfoMaker reports in other applications using OLE 2.0.

There are two ways to embed an InfoMaker report in an application. You can:

- ◆ Create a new InfoMaker report from within another application
- ◆ Add a report saved as a Powersoft report (PSR) file

### Creating a new InfoMaker report from within another application

You can create a new InfoMaker report from within any application that works with OLE 2.0.

When a report embedded in an application is active, the InfoMaker toolbar displays instead of the application's toolbar. This allows you to work with the report in place. When you return to the application, the application's toolbar displays. You can activate the report at any time by double-clicking it.

---

#### **Report size limited to one page**

When you embed a Powersoft report, you will see one page of information. If the report is long, the information will be too tiny to be legible. This is a limitation of OLE. You should plan the size of the report to fit easily on one page.

---

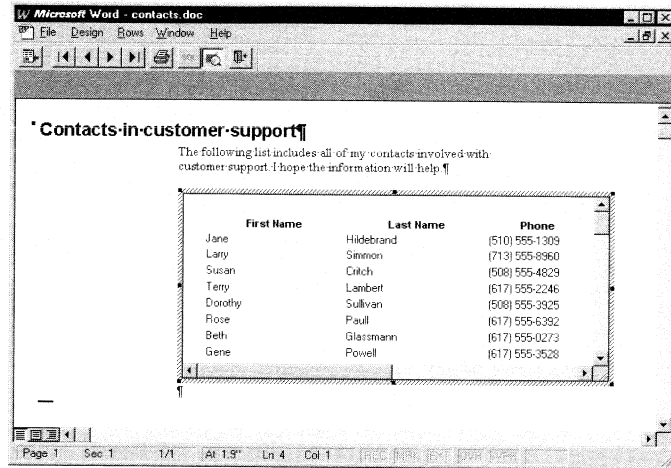
The following procedure shows how to create a report in a Word 6 or Word 7 document.

#### ❖ **To create a new InfoMaker report from within Word:**

- 1 Select Insert>Object from the Word menu bar.  
The Object dialog box displays.
- 2 Click the Create New tab (if it's not on top) and select Powersoft Report.  
You may need to scroll the list of object types.
- 3 Click OK.  
The InfoMaker New Report dialog box displays.

- 4 Select a data source and presentation style, click OK, and provide information for the new report.

The report displays in preview mode and the InfoMaker menu and toolbar display:



You can edit the report just as you would any report. Because the report is a metafile when it is inactive, you can scale it and add a border using Word options.

## Adding a report saved as a PSR file

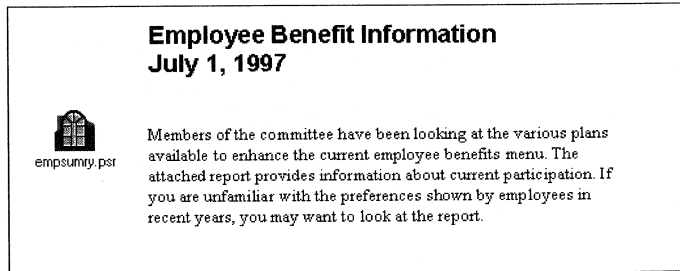
You can add a report previously saved as a PSR file from within any application that works with OLE 2.0. Optionally, you can choose to display the report as an icon. To access the iconized report and start InfoMaker, you double-click the icon.

### ❖ To add a report saved as a PSR file from within Word:

- 1 Select Insert>Object from the menu bar.  
The Object dialog box displays.
- 2 Click the Create from File tab and select the PSR file.  
Optionally, select the Display as Icon checkbox.

3 Click OK.

The Object dialog box closes. An icon representing the report displays at the insertion point:



To view the report and print it, double-click the icon.

# Data Pipelines

This part introduces the Data Pipeline painter and describes how to use it to pipe data.

---

**Access to the Data Pipeline painter**

To have access to the Data Pipeline painter, install InfoMaker's pipeline component.

---



About this chapter

This chapter describes how to use the Data Pipeline painter to create data pipelines, which let you reproduce database data in various ways.

Contents

<b>Topic</b>	<b>Page</b>
About data pipelines	500
Creating a data pipeline	503
Modifying the data pipeline definition	506
Correcting pipeline errors	516
Saving a pipeline	518
Using an existing pipeline	519
Pipeline examples	520

## About data pipelines

The Data Pipeline painter gives you the ability to quickly reproduce data within a database, across databases, or even across DBMSs. To do that, you create a data pipeline which, when executed, **pipes** the data as specified in the definition of the data pipeline.

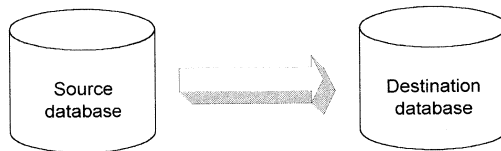
### What you can do

With the Data Pipeline painter, you can perform some tasks that would otherwise be very time consuming. For example, you can:

- ◆ Pipe data (and extended attributes) from one or more tables to a table in the same DBMS or a different DBMS
- ◆ Pipe an entire database, a table at a time, to another DBMS (and if needed, pipe the database's repository tables)
- ◆ Create a table with the same design as an existing table but with no data
- ◆ Pipe corporate data from a database server to a SQL Anywhere database on your computer so you can work on the data and report on it without needing access to the network
- ◆ Upload local data that changes daily to a corporate database
- ◆ Create a new table when a change (such as allowing or disallowing NULLs or changing primary key or index assignments) is disallowed in the Table painter

### Source and destination databases

You can use the Data Pipeline painter to pipe data from *one or more* tables in a **source database** to *one* table in a **destination database**:



You can pipe all data or selected data in one or more tables. For example, you can pipe a few columns of data from one table or data selected from a multitable join. You can also pipe a view.

When you pipe data, the data in the source database remains in the source database and is reproduced in a new or existing table in the destination database.

Although the source and destination can be the same database, they are usually different ones, and they can even have different DBMSs. For example, you can pipe data from a SQL Server database to a SQL Anywhere database on your computer.



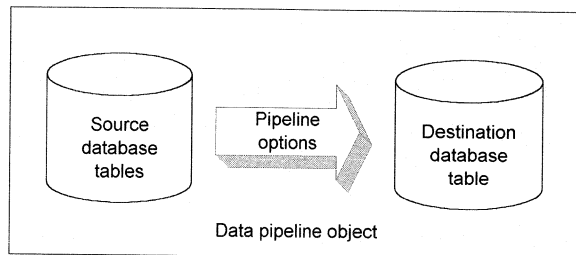
## Defining a data pipeline

When you use the Data Pipeline painter to create a pipeline, you define:

- ◆ Source database
- ◆ Destination database
- ◆ Source tables to access and the data to retrieve from them
- ◆ Pipeline operation
- ◆ Destination table

Once you define it

After you create a pipeline, you can execute it immediately. If you want, you can also save it as a named object to use and reuse:



Saving a pipeline enables you to pipe the data that may have changed since the last pipeline execution or to pipe the data to other databases later.

Data type support

Each DBMS has certain data types it supports. When you pipe data from one DBMS to a different DBMS, InfoMaker makes its best guess for the destination data types. You can correct InfoMaker's best guess in your pipeline definition as needed.

The Data Pipeline painter supports the piping of columns of any data type, including columns with blob data.

**FOR INFO** For information about piping a column that has a blob data type, see "Piping blob data" on page 513.

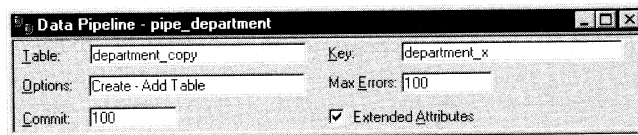
## Piping extended attributes

The first time PowerBuilder or InfoMaker connects to a database, it creates five Powersoft system tables, called the **Powersoft repository**. These system tables initially contain default extended attribute information for tables and columns. In PowerBuilder or InfoMaker, you can create extended attribute definitions, such as column headers and labels, edit styles, display formats, and validation rules.

**FOR INFO** For more information about the repository, see the Appendix, "The Powersoft Repository".

Piping extended attributes automatically

When you pipe data, you can specify that you want to pipe the extended attributes associated with the columns you are piping. You do this by selecting the Extended Attributes checkbox in the Data Pipeline painter workspace:



When the Extended Attributes checkbox is selected, the extended attributes associated with the source database's selected columns automatically go into the repository of the destination database, with the following exception.

When extended attributes are not automatically piped

When you pipe data that has an edit style, display format, or validation rule associated with it, the style, rule, or format will *not* be piped if one with the same name exists in the repository of the destination database. In this situation, the data uses the style, rule, or format already present in the destination database.

For example, for the Phone column in the Employee table, the display format with the name Phone\_format would be piped unless a display format with the name Phone\_format already exists in the destination database. If such a display format exists, the Phone column would use the Phone\_format display format in the destination database.

Piping the Powersoft repository

Selecting the Extended Attributes checkbox *never* results in the piping of named display formats, edit styles, and validation rules that are stored in the repository but independent of data. If you want such extended attribute definitions from one database to exist in another database, you can pipe the appropriate Powersoft repository table or a selected row or rows from the table.

If you want to reproduce an entire database, you can pipe all database tables and repository tables, one table at a time.

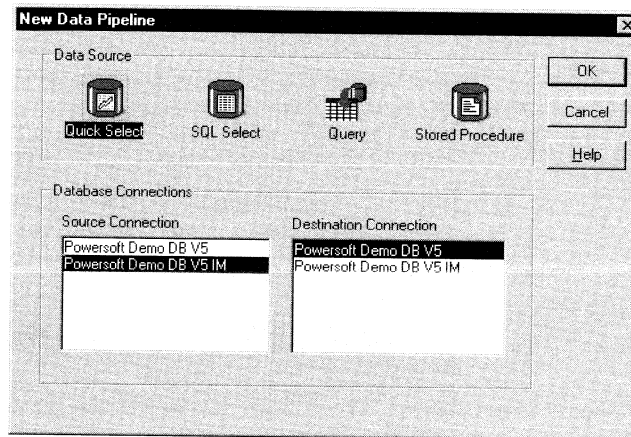
## Creating a data pipeline

You have a number of choices when creating a data pipeline. This section walks you through them.

❖ **To create a data pipeline:**

- 1 Click the Pipeline button in the PowerBar.  
The Select Data Pipeline dialog box displays.
- 2 Click New.

The New Data Pipeline dialog box displays:



The Source Connection and Destination Connection boxes display database profiles that have been defined. The last database you connected to is selected as the source. The first database on the destination list is selected as the destination.

---

### **If you don't see the connections you need**

To create a pipeline, the databases you want to use for your source and destination must each have a database profile defined. If you don't see profiles for the databases you want to use, cancel the New Data Pipeline dialog box, then define those profiles.

**FOR INFO** For information about defining profiles, see "Changing the destination and source databases" on page 514.

---

- 3 Select a data source.

The data source determines how InfoMaker retrieves data when you execute a pipeline:

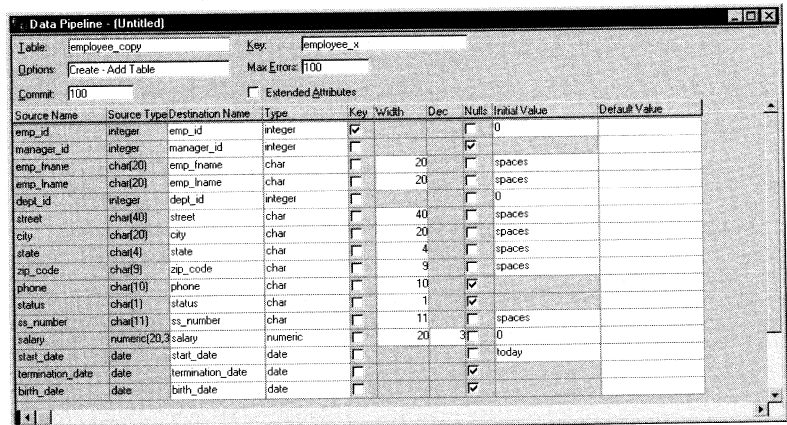
Data source	Use it if
Quick Select	The data is from tables that are connected through a key and you only need to sort and limit data
SQL Select	You want more control over the SQL SELECT statement generated for the data source or your data is from tables that are not connected through a key
Query	The data has been defined as a query
Stored Procedure	The data is defined in a stored procedure

- 4 Select the source and destination connections and click OK.
- 5 Define the data to pipe.

How you do this depends on what data source you chose in step 3.

FOR INFO For complete information about using each data source and defining the data, see Chapter 4, "Defining Reports".

When you finish defining the data to pipe, the Data Pipeline painter workspace displays the pipeline definition, which includes a pipeline operation, a checkbox for specifying whether to pipe extended attributes, and source and destination items:



The pipeline definition is InfoMaker's best guess based on the source data you specified.

- 6 Modify the pipeline definition as needed.

FOR INFO For information, see "Modifying the data pipeline definition" next.

7 (Optional) Modify the source data as needed:

- ◆ Click the Edit SQL button.

*or*

Select Design>Edit Data Source from the menu bar.

**FOR INFO** For information about working in the Select painter, see Chapter 4, "Defining Reports".

When you return to the Data Pipeline painter workspace, InfoMaker reminds you that the pipeline definition will change. Click OK to accept the definition change.

8 If you want to try the pipeline now, click the Execute button.

*or*

Select Design>Execute from the menu bar.

InfoMaker retrieves the source data and executes the pipeline. If you specified retrieval arguments in the Select painter, InfoMaker first prompts you to supply them.

---

### **The role of the Query governor**

Options you set in the Query Governor affect the Data Pipeline painter when you specify the data and when data is piped. In the Query Governor, you can set data selection options, and limit the number of rows piped and the elapsed piping time.

**FOR INFO** For more information, see Chapter 2, "Managing Your Environment".

---

During execution, the number of rows read and written, the elapsed execution time, and the number of errors display in MicroHelp. You can stop execution yourself or InfoMaker may stop execution when errors occur.

**FOR INFO** For information about execution and how rows are committed to the destination table, see "When execution stops" on page 511.

9 Save the pipeline definition if appropriate.

**FOR INFO** For information, see "Saving a pipeline" on page 518.

---

### **Seeing the results of piping data**

You can see the results of piping data by connecting to the destination database and opening the destination table.

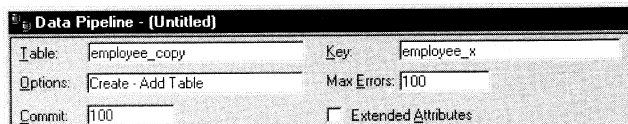
---

## Modifying the data pipeline definition

After you create a pipeline definition, you can modify it in a variety of ways. This depends on what pipeline operation you select, the destination DBMS, and what you are trying to accomplish by executing the pipeline.

### Items you can modify in the workspace

**At the top of the workspace** These items apply to the destination table:



Item	Description	Default	How to edit
Table	Name of the destination table	If source and destination are different, name of first table specified in the source data or name of the stored procedure; if the same, <i>_copy</i> is appended	For Create or Replace, enter a name For Refresh, Append, or Update, select a name from the dropdown listbox
Options	Pipeline operation: Create, Replace, Refresh, Append, or Update	Create - Add Table	Select an option from the dropdown listbox
Commit	Number of rows piped to the destination database before InfoMaker commits the rows to the database	100 rows	Select a number, <i>All</i> , or <i>None</i> from the dropdown listbox
Key	Key name for the table in the destination database	If the source is only one table, the table name is followed by <i>_x</i>	For Create or Replace, enter a name. Not editable for other pipeline operations
Max Errors	Number of errors allowed before the pipeline stops	100 errors	Select a number or <i>No Limit</i> from the dropdown listbox

Item	Description	Default	How to edit
Extended Attributes	For Create and Replace, a checkbox that specifies whether or not the extended attributes of the selected source columns are piped to the repository of the destination database. Does not display for Refresh, Append, or Update	Not checked	Click the checkbox

**At the bottom left of the workspace** These items show the source column names and data types. They are not editable, because you specified them as the data source:

Source Name	Source Type	Destination Name	Type	Key	Width	Dec	Nulls	Initial Value	Default Value
emp_id	integer	emp_id	integer	<input checked="" type="checkbox"/>			<input type="checkbox"/>	0	
manager_id	integer	manager_id	integer	<input type="checkbox"/>			<input checked="" type="checkbox"/>		
emp_fname	char(20)	emp_fname	char	<input type="checkbox"/>	20		<input type="checkbox"/>	spaces	
emp_lname	char(20)	emp_lname	char	<input type="checkbox"/>	20		<input type="checkbox"/>	spaces	
dept_id	integer	dept_id	integer	<input type="checkbox"/>			<input type="checkbox"/>	0	
street	char(40)	street	char	<input type="checkbox"/>	40		<input type="checkbox"/>	spaces	
city	char(20)	city	char	<input type="checkbox"/>	20		<input type="checkbox"/>	spaces	
state	char(4)	state	char	<input type="checkbox"/>	4		<input type="checkbox"/>	spaces	
zip_code	char(3)	zip_code	char	<input type="checkbox"/>	9		<input type="checkbox"/>	spaces	
phone	char(10)	phone	char	<input type="checkbox"/>	10		<input checked="" type="checkbox"/>		
status	char(1)	status	char	<input type="checkbox"/>	1		<input checked="" type="checkbox"/>		
ss_number	char(11)	ss_number	char	<input type="checkbox"/>	11		<input type="checkbox"/>	spaces	
salary	numeric(20,3)	salary	numeric	<input type="checkbox"/>	20	3	<input type="checkbox"/>	0	
start_date	date	start_date	date	<input type="checkbox"/>			<input type="checkbox"/>	today	
termination_date	date	termination_date	date	<input type="checkbox"/>			<input checked="" type="checkbox"/>		
birth_date	date	birth_date	date	<input type="checkbox"/>			<input checked="" type="checkbox"/>		

**At the bottom right of the workspace** These items apply to the destination table's columns and key. They are mostly editable only for the Create and Replace pipeline operations:

Item	Description	Default	How to edit
Destination Name	Column name	Source column name	Enter a name
Type	Column data type	If the DBMS is unchanged, source column data type. If the DBMS is different, a best-guess data type	Select a type from the dropdown listbox
Key	Whether the column is a key column (check means yes)	Source table's key columns (if the source is only one table and all key columns are selected)	Select or clear checkboxes
Width	Column width	Source column width	Enter a number
Dec	Decimal places for the column	Source column decimal places	Enter a number

Item	Description	Default	How to edit
Nulls	Whether NULL is allowed for the column (check means yes)	Source column value	Select or clear checkboxes
Initial Value	Column initial value	Source column initial value (if no initial value, character columns default to <i>spaces</i> and numeric columns default to 0)	Select an initial value from the dropdown listbox
Default Value	Column default value	Source column default value that's assigned in the database	Select a default value from the dropdown listbox or enter a default value. Keyword values depend on destination DBMS

## Choosing a pipeline operation

When InfoMaker pipes data, what happens in the destination database depends on which pipeline operation you choose in the Options dropdown listbox:

When you choose this pipeline operation	This happens in the destination database
Create - Add Table	A new table is created and rows selected from the source tables are inserted  If a table with the specified name already exists in the destination database, a message displays and you must select another option or change the table name
Replace - Drop/Add Table	An existing table with the specified table name is dropped, a new table is created, and rows selected from the source tables are inserted  If no table exists with the specified name, a table is created
Refresh - Delete/Insert Rows	All rows of data in an existing table are deleted, and rows selected from the source tables are inserted
Append - Insert Rows	All rows of data in an existing table are preserved, and new rows selected from the source tables are inserted



When you choose this pipeline operation	This happens in the destination database
Update - Update/Insert Rows	Rows in an existing table that match the key criteria values in the rows selected from the source tables are updated, and rows that don't match the key criteria values are inserted

## Dependency of modifications on pipeline operation

The modifications you can make in the workspace depend on the pipeline operation you have chosen.

### When using Create - Add Table or Replace - Drop/Add Table

When you select the Create option (the default) or the Replace option, you can change these items:

You can	Comment
Change the destination table definition	Follow the rules of the destination DBMS
Have both a key name and key columns or neither	Specify key columns by selecting one or more checkboxes to define a unique identifier for rows
Allow or disallow NULL for a column	If NULL is allowed (checkbox selected), no initial value is allowed If NULL is not allowed, an initial value is required. The words <i>spaces</i> (a string filled with spaces) and <i>today</i> (today's date) are initial value keywords
Modify the Commit and Max Errors values	
Specify an initial value and a default value	

If you have specified key columns and a key name and if the destination DBMS supports primary keys, the Data Pipeline painter creates a primary key for the destination table. If the destination DBMS does not support primary keys, a unique index is created.

---

### **For Oracle databases**

InfoMaker generates a unique index for Oracle databases.

---

If you try to use the Create option, but a table with the specified name already exists in the destination database, InfoMaker tells you and you must select another option or change the table name.

When you use the Replace option, InfoMaker warns you that you are deleting a table, and you can choose another option if needed.

### **When using Refresh - Delete/Insert Rows or Append - Insert Rows**

For the Refresh and Append options, the destination table already exists. You can:

- ◆ Select an existing table from the dropdown listbox in the Table box
- ◆ Modify the Commit and Max Errors values
- ◆ Change the initial value for a column

### **When using Update - Update/Insert Rows**

For the Update option, the destination table already exists. You can:

- ◆ Select an existing table from the dropdown listbox in the Table box
- ◆ Modify the Commit and Max Errors values
- ◆ Change the Key columns in the destination table's key (primary key or unique index, depending on what the DBMS supports), but key columns must be selected

The key determines the UPDATE statement's WHERE clause.

- ◆ Change the initial value for a column

---

### **Bind variables and the Update option**

If the destination database supports bind variables, the Update option takes advantage of them to optimize pipeline execution.

---

## When execution stops

Execution of a pipeline can stop for any of these reasons:

- ◆ You click the Cancel button
  - During the execution of a pipeline, the Execute button in the PainterBar changes to the Cancel button.
- ◆ The error limit is reached
- ◆ The Query Governor's row or elapsed-time limit is reached

If there are rows that can't be piped to the destination table for some reason, those error rows display once execution stops. You can correct error rows or return to the workspace to change the pipeline definition and then execute it again.

FOR INFO For information, see "Correcting pipeline errors" on page 516.

## Whether rows are committed

When rows are piped to the destination table, they are first inserted and then either committed or rolled back. Whether rows are committed depends on:

- ◆ The Commit and Max Errors values
- ◆ When errors occur during execution
- ◆ Whether you click the Cancel button or InfoMaker stops execution

When you stop  
execution

If the Commit value is	Then when you click Cancel or a Query Governor limit is reached
A number <i>n</i>	Each row that was piped is committed
<i>All or None</i>	Each row that was piped is rolled back

For example, if you click the Cancel button when the 24th row is piped and if the Commit value is 20, then:

- ◆ 20 rows are piped and committed
- ◆ 3 rows are piped and committed
- ◆ Piping stops

If the Commit value were *All or None*, 23 rows would be rolled back.

When InfoMaker stops execution

<b>If the Commit value is</b>	<b>And the Max Errors value is</b>	<b>Then when InfoMaker stops execution because the error limit is reached</b>
A number <i>n</i>	<i>No limit</i> or a number <i>m</i>	Rows are piped and committed <i>n</i> rows at a time until the Max Errors value is reached
<i>All</i> or <i>None</i>	<i>No limit</i>	Each row that pipes without error is committed
<i>All</i> or <i>None</i>	A number <i>n</i>	If the number of errors is less than <i>n</i> , all rows are committed  If the number of errors is equal to <i>n</i> , each row that was piped is rolled back. No changes are made

For example, if an error occurs when the 24th row is piped and the Commit value is 10 and the Max Errors value is 1, then:

- ◆ 10 rows are piped and committed
- ◆ 10 rows are piped and committed
- ◆ 3 rows are piped and committed
- ◆ Piping stops

If the Commit value were *All* or *None*, 23 rows would be rolled back.

## About transactions

A transaction is a logical unit of work done by a DBMS, within which either all the work in the unit must get done or none of the work in the unit must get done. If the destination DBMS doesn't support transactions or isn't connected in transaction mode, each row that is inserted or updated is committed.

## About the All and None commit values

In the Data Pipeline painter, the Commit values *All* and *None* have the same meaning.

## Piping blob data

Blob data is data having a data type that specifies the data as *binary large-objects* such as a Microsoft Word documents or Excel spreadsheets. A data pipeline can pipe columns containing blob data.

The name of the data type that supports blob data varies by DBMS, for example:

DBMS	Data types that support blob data
SQL Anywhere	LONG BINARY LONG VARCHAR (if more than 32 KB)
Sybase SQL Server	IMAGE TEXT
Microsoft SQL Server	IMAGE TEXT
Oracle	RAW LONG RAW
Informix	BYTE TEXT

**FOR INFO** For information about the data type that supports blob data in your DBMS, see your DBMS documentation.

Adding blob columns to a pipeline definition

When you select data to pipe, you cannot select a blob column as part of the data source because blobs can't be handled in a `SELECT` statement. After the pipeline definition is created, you add blob columns, one at a time, to the definition.

### ❖ To add a blob column to a pipeline definition:

- 1 Select `Design>Database Blob` from the menu bar.

---

#### **If the Database Blob menu item is grayed**

The Database Blob menu item is grayed if the pipeline definition doesn't contain a unique key for at least one source table, or if the pipeline operation is Refresh, Append, or Update and the destination table has no blob columns or you've associated all destination blob columns with source blob columns.

---

The Database Binary/Text Large Object property sheet displays.

The Table box has a dropdown list of tables in the source database that have a primary key and contain blob columns.

- 2 In the Table box, select the table that contains the blob column you want to add to the pipeline definition.

For example, in the Powersoft Demo Database, the ole table contains a blob column named Object with the large binary data type.

- 3 In the Large Binary/Text Column box, select a column that has a blob data type.
- 4 In the Destination Column box, change the name of the destination column for the blob if you want.  
If you want to add the column and see changes you make without closing the dialog box, click Apply after each change.
- 5 When you have specified the blob source and destination as needed, click OK

❖ **To edit the source or destination name of the blob column in the pipeline definition:**

- ◆ Display the blob column's popup menu and select Properties.

❖ **To delete a blob column from the pipeline definition:**

- ◆ Display the blob column's popup menu and select Clear.

Executing a pipeline with blob columns

After you've completed the pipeline definition by adding one or more blob columns, you can execute the pipeline. When you do, rows are piped a block at a time, depending on the Commit value. For a given block, Row 1 is inserted, then Row 1 is updated with Blob 1, then Row 1 is updated with Blob 2, and so on. Then Row 2 is inserted, and so on until the block is complete.

If a row is not successfully piped, the blob is not piped. Blob errors display, but the blob itself does not display. When you correct a row in error and execute the pipeline, the pipeline pipes the blob.

## Changing the destination and source databases

Changing the destination

When you create a pipeline, you can change the destination database. And if you want to pipe the same data to more than one destination, you can change the destination database again and re-execute.

❖ **To change the destination database:**

- ◆ Click the Destination button in the PainterBar.  
*or*  
Select File>Destination Connect from the menu bar.

Changing the source

Normally you don't change the source database, because your pipeline definition is dependent on it. But if you need to (maybe because you are no longer connected to that source) you can.

❖ **To change the source database:**

- ◆ Select File>Source Connect from the menu bar.

Working with database profiles

At any time in the Data Pipeline painter, you can edit an existing database profile or create a new one.

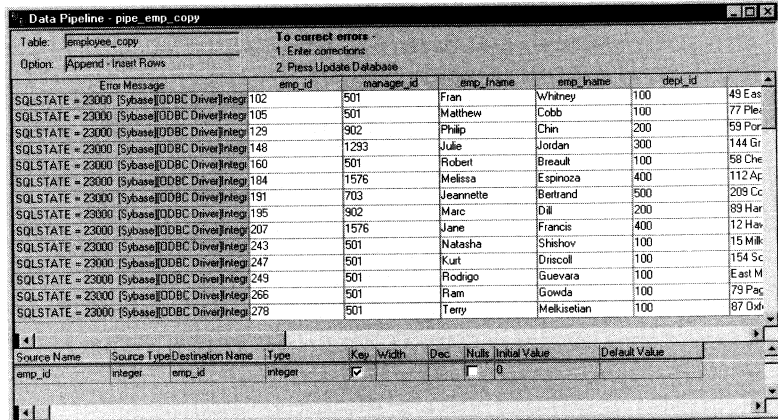
❖ **To edit or create a database profile:**

- ◆ Click the Database Profile button in the PainterBar and then click the Edit button or the New button.

FOR INFO For information about how to edit or define a database profile, see *Connecting to Your Database*.

## Correcting pipeline errors

If the pipeline can't pipe certain rows to the destination table for some reason, InfoMaker displays the error rows:



What you see

InfoMaker shows:

- ◆ Name of the table in the destination database
- ◆ Pipeline operation you chose in the Option box
- ◆ Error messages to identify the problem with each row
- ◆ Data values in the error rows
- ◆ Source and destination column information

What you can do

You can correct the error rows by changing one or more of their column values so the destination table will accept them; or you can ignore the error rows and return to the Data Pipeline painter workspace. If you return to the workspace, you cannot redisplay the error rows without re-executing the pipeline.

### Before you return to the workspace

You may want to print the list of errors or save them in a file. Select File>Print or File>Save As from the menu bar.

- ❖ **To return to the Data Pipeline painter workspace without correcting errors:**
  - ◆ Click the Design button.



❖ **To correct pipeline errors:**

- 1 Change data values for the appropriate columns in the error rows.
- 2 Click the Update DB button.

*or*

Select Design>Update Database from the menu bar.

InfoMaker pipes rows in which errors were corrected to the destination table and displays any remaining errors.

- 3 Repeat steps 1 and 2 until all errors are corrected.

The Data Pipeline painter workspace displays.

Viewing an error message

Sometimes you cannot see an entire error message because the column isn't wide enough:

Error Message	emp_id	emp_fname	emp_lname	dept_name	salary
SQLSTATE = 23000 [WATCOM]ODBC Driver[1W/102		Fran	Whitney	R & D	45700.000
SQLSTATE = 23000 [WATCOM]ODBC Driver[1W/103		Mark	Morris	R & D	20000.000
SQLSTATE = 23000 [WATCOM]ODBC Driver[1W/104		Rolln	Ovelbey	R & D	20000.000

❖ **To view an error message**

- ◆ Move the pointer to the error message and press the RIGHT ARROW key to scroll through it.

*or*

Drag the Error Message column border to the width needed:

Error Message	emp_id
SQLSTATE = 23000 [WATCOM]ODBC Driver[1WATCOM SQL Integrity constraint violation: primary key for table 'employee' is not unique	102
SQLSTATE = 23000 [WATCOM]ODBC Driver[1WATCOM SQL Integrity constraint violation: primary key for table 'employee' is not unique	103
SQLSTATE = 23000 [WATCOM]ODBC Driver[1WATCOM SQL Integrity constraint violation: primary key for table 'employee' is not unique	104

**Making the error messages shorter**

For ODBC data sources, you can set the DBParm MsgTerse parameter in the destination database profile to make the error messages shorter. If you type:

```
MsgTerse = 'Yes'
```

then the SQLSTATE error number won't appear.

FOR INFO For more information, see *Connecting to Your Database*.

## Saving a pipeline

When you have generated a pipeline definition in the Data Pipeline painter workspace, you should save the pipeline. You can then reuse it later.

❖ **To save a pipeline:**

- ◆ Click the Save button  
*or*  
Select File>Save from the menu bar.

For a new pipeline

When you save a pipeline for the first time:

- ◆ Specify a name (any valid identifier up to 40 characters).

InfoMaker saves the pipeline in the current library (specified in the Environment painter).

**FOR INFO** For information about changing the current library, see Chapter 2, "Managing Your Environment".

## Using an existing pipeline

If you save a pipeline, you can modify and execute it any time. You can also pipe data that may have changed since the last pipeline execution or pipe data to other databases.

❖ **To use an existing pipeline:**

- 1 Click the Pipeline button in the PowerBar.

The Select Data Pipeline dialog box displays. Pipelines in the current library (PBL) are listed. If you do not see the pipeline you want, close the Data Pipeline painter, open the Environment painter, and change the current library.

- 2 Select the pipeline you want to execute.

- 3 Click the Design button.

The Data Pipeline painter workspace displays.

- 4 If you want to change the pipeline operation, select a new option.

- 5 Modify the pipeline definition as needed.

- 6 Execute and/or save the pipeline.

## Pipeline examples

Updating data in a destination table

You may want to pipe data and then update the data often.

❖ **To update a destination table:**

- 1 Click the Pipeline button, select an existing pipeline that you executed before, and click OK.

The pipeline definition displays. Since this pipeline has been executed before, the table exists in the destination database.

- 2 Select the Update option in the pipeline definition.
- 3 Execute the pipeline.

The destination table is updated with current data from the source database.

Reproducing a table definition with no data

You can force a pipeline to create a table definition and not pipe data. It's easiest to do this using SQL Select as the data source.

❖ **To reproduce a table definition with no data:**

- 1 Click the Pipeline button, click New, select SQL Select as the data source and specify the source and destination databases, and click OK.
- 2 In the Select painter, open the table you want to reproduce and select all columns.
- 3 In the Where tab, type an expression that will never evaluate to true, such as  $1 = 2$ .
- 4 Click the SQL Select button to create the pipeline definition.
- 5 Select the Extended Attributes checkbox.
- 6 Click the Execute button to execute the pipeline.

The table definition is piped to the destination database, but no rows of data are piped. You can open the new table in the Database painter and then click the Grid, Table, or Freeform button to view the data. As specified, there is no data.

**If you don't use SQL Select** If you use a data source other than SQL Select, you can reproduce a table definition by creating the pipeline definition, editing the data source (which opens the Select painter), and then typing the expression that will never evaluate to true in the Where tab.

Piping a table to many databases

In the Data Pipeline painter workspace, you can execute a pipeline many times with a different destination database each time.

❖ **To pipe a table to many databases:**

- 1 Select File>Destination Connect from the menu bar to change the destination to the database you want.
- 2 Execute the pipeline.
- 3 Repeat steps 1 and 2 for each database you want.



# Forms

This part describes how to use forms to display and change information in your database

---

**Access to the Form painter**

To have access to the Form painter, install InfoMaker's form component.

---





**About this chapter**

You use forms to add data to your database easily and efficiently. This chapter introduces InfoMaker forms and provides basic information about working with forms.

**Contents**

<b>Topic</b>	<b>Page</b>
About forms	526
Creating and saving forms	536
Working with forms	546
Controlling updates	555

## About forms

An InfoMaker **form** is an electronic document you use to enter data in a database. The form displays existing data from your database. You can change the existing data and add new data. You can also print a form. Each form has procedures associated with it, which include common database tasks such as delete, insert, and update.

Why use a form for data entry

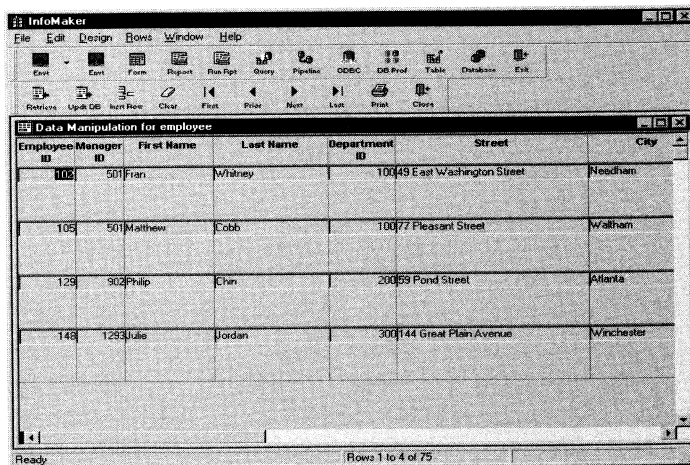
In InfoMaker, you can enter data in your database in two ways:

- ◆ In the Database painter, you can preview a table, change data or insert a new row of data, and update the database
- ◆ In the Form painter, you can run a form, change data or insert a new row of data, and update the database

If you need to add or change data in your database frequently, making changes directly in the database tables is inefficient and time-consuming. A form is designed to make data entry easier and faster.

You can design a form so you can see all columns of data on your screen. Your design can make data easy to view on the screen and easy to read if printed. And you can create edit styles and display formats to make it easier to enter and view data.

Here's the Employee table in the Powersoft Demo Database, which has 20 columns of data for 75 employees. When you view it in the Data Manipulation painter, you can see only a few rows and columns of data at a time:



The screenshot shows the InfoMaker Data Manipulation painter window. The title bar reads "Data Manipulation for employee". The window contains a table with the following columns: Employee ID, Manager ID, First Name, Last Name, Department ID, Street, and City. The data is as follows:

Employee ID	Manager ID	First Name	Last Name	Department ID	Street	City
102	501	Fran	Whitney	101	49 East Washington Street	Needham
105	501	Matthew	Cobb	100	77 Pleasant Street	Waltham
129	902	Philip	Chen	200	69 Pond Street	Atlanta
148	1293	Jake	Jordan	300	144 Great Plain Avenue	Winchester

The status bar at the bottom of the window indicates "Ready" and "Rows 1 to 4 of 75".

Changing data or entering data for a new employee directly in the Employee table is difficult. To see other columns you need to scroll (or InfoMaker scrolls for you as you tab from column to column), and you can never see all the data for an employee at one time.

Instead, you could use the following form to enter or change employee data. The form uses InfoMaker's Freeform form style. When you run the form, here's what it looks like:

The screenshot shows the InfoMaker application window with a menu bar (File, Edit, Rows, Window, Help) and a toolbar. The main window is titled "Employee Data" and contains a form with the following fields and values:

Employee ID:	100	Birth Date:	06/06/1958
Manager ID:	501	Soc. Sec. No.:	017-34-9033
Emp. First Name:	Fran	Salary:	\$45,700.00
Emp. Last Name:	Whitney	Start Date:	02/26/1987
Department ID:	100	Termination Date:	00/00/0000
Street:	49 East Washington Street		
City:	Needham		
State:	MA		
Zip Code:	02192		
Phone:	(617) 555-3985		
Sex:	<input type="radio"/> Male <input checked="" type="radio"/> Female		
Health Insurance:	<input checked="" type="checkbox"/>		
Life Insurance:	<input checked="" type="checkbox"/>		
Day Care:	<input type="checkbox"/>		
Status:	<input checked="" type="radio"/> Active <input type="radio"/> Terminated <input type="radio"/> On Leave		

You can use this form to change data easily. And you can insert data for a new employee: click the Insert button and a blank form displays. You can see all the data for each employee at one time, and with additional enhancements, the presentation of the data can be improved for printing.

### Forms in InfoMaker and in an InfoMaker application

After you design a form, you can use it within InfoMaker. You can also take the form, package it in an InfoMaker application with other forms and reports you've created, and distribute your application.

For example, you could create an Employee Data application that includes the Employee Data form and many employee reports that are important to your organization. When users run the application, they get up-to-date reports and they can update the Employee database.

FOR INFO For information about creating InfoMaker applications, see Chapter 17, "Working with Applications".

## About the Form painter's PainterBar

The Form painter's PainterBar provides buttons for performing common activities you might do when creating or altering a form definition. There's also a menu item corresponding to most buttons.

## About creating new forms

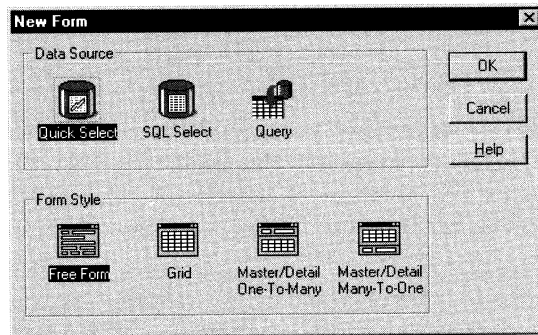
When you create a new form with InfoMaker, you always specify the data source and form style of the form. Then you define the data for the form, and InfoMaker generates the basic form.

Once you have the basic form, you can name the form and save it. Then you can continue to enhance the form to make it easier to use.

The following discussion introduces you to a few form concepts and terms. When you are ready to create a particular type of form, you can follow the steps for the type.

### New Form dialog box

You specify the data source and form style in the New Form dialog box:



### Data sources

The data source you use determines how InfoMaker retrieves data for your form. You can select one of three data sources when you create a form:

<b>Data source</b>	<b>Pick this data source when</b>
Quick Select	The data is from tables that are connected through a key, and you only need to sort and limit data
SQL Select	The data is from tables that are not connected through a key, or you want more control over the SQL SELECT statement generated for the data source
Query	The data has been defined as a query

**For master/detail forms**

You always use the Quick Select data source for master/detail forms.

**FOR INFO** For complete information about data sources, see "Defining the data source" on page 155.

## Form styles

A **form style** is a predefined way of presenting and processing information in a form. Each form you build in InfoMaker is based on an existing form style.

InfoMaker provides four built-in form styles:

<b>Form</b>	<b>What it shows</b>
Freeform	One row of data at a time
Grid	Rows of data in a grid
Master/Detail One-To-Many	One item in a freeform area at the top and a grid with details about the item at the bottom
Master/Detail Many-To-One	Many items in a grid at the top and a freeform area with details about an item at the bottom

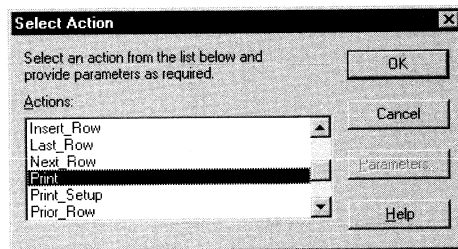
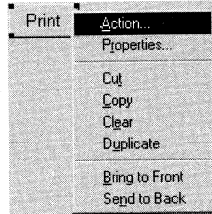
In addition to the built-in form styles, PowerBuilder developers in your organization can develop custom form styles to meet your organization's needs. Custom form styles also display in the New Form dialog box.

**FOR INFO** For information about creating custom form styles, talk to PowerBuilder developers in your organization.

## Actions in a form style

Each form style has a set of predefined actions. An **action** is a procedure you can attach to buttons you place in a form. Usually a style includes the common database actions (insert, update, and delete rows). Each of InfoMaker's built-in form styles has these database actions and additional actions that are useful and appropriate to the form style.

For example, you can place a button in a form and then attach the Print action to it. After you run the form, you can print the current data by clicking the button:



Actions are also available through the toolbar and menu items when you run a form.

## Freeform forms

You use freeform forms for basic data maintenance.

In freeform forms, you see one row of data at a time. You can arrange the information any way you want. When you run the form, you can add, modify, and delete rows of data.

For example, the following form is a freeform form that allows you to view and update customer information. This form uses all columns in the Customer table in the Powersoft Demo Database.

After a few enhancements have been made to the basic form, here's the freeform form with data after running it from the workspace:

The screenshot shows a window titled "Customer Maintenance". On the left is a logo with a stylized 'C' and 'M' and the date "3/20/96". The main area contains two sections:

**Customer Information**

- Customer ID: 101
- Company Name: The Power Group
- Address: 3114 Pioneer Avenue
- City, State, Zip: Rutherford NJ 07070

**Contact Information**

- Contact: Michaels Devlin
- Phone: (201) 555-8966

## Grid forms

You use grid forms for basic data maintenance where you want to be able to view and update more than one row of data at a time.

The data in a grid form displays in a rigid grid. When running a grid form, you can resize and reorder columns.

For example, the following form is a grid form that allows you to view information for many customers at a time. This form uses all columns in the Customer table in the Powersoft Demo Database:

The screenshot shows a window titled "Untitled" containing a grid with the following data:

Customer ID	First Name	Last Name	Address	City	State	Zip Code
101	Michaels	Devlin	3114 Pioneer Avenue	Rutherford	NJ	07070
102	Beth	Feiser	1033 Whippary Road	New York	NY	10154
103	Enn	Niedringhaus	1990 Windsor Street	Faak	PA	19301
104	Meghan	Mason	650 Dundas Street East	Knowville	TN	37513
105	Laura	McCarthy	1210 Highway 36	Camel	IN	46032
106	Paul	Phillips	2000 Cherry Creek N. Dr.	Middletown	CT	06459
107	Kelly	Colburn	18131 Valico Parkway	Raleigh	NC	27695
108	Matthew	Goforth	11801 Wayzata Blvd.	Chattanooga	TN	37421

## Working in a grid form

When you design and run a grid form, you can resize and reorder columns.

❖ **To resize a column:**

- 1 Position the pointer at a column boundary.  
The pointer changes shape to a 2-headed arrow.
- 2 Drag the mouse to move the boundary.
- 3 Release the mouse button when the column is the correct width.

❖ **To reorder columns:**

- 1 Select a column heading.

InfoMaker selects the column and displays a line representing the column border:

Company Name	Address	City
The Power Group	3114 Pioneer Avenue	Rutherford
AMF Corp.	1033 Whippany Road	New York
Darling Associates	1990 Windsor Street	Paoli
P.S.C.	550 Dundas Street East	Knoxville
Arno & Sons	1210 Highway 36	Carmel
Ralston Inc.	2000 Cherry Creek N.	Middletown
The Home Club	18131 Vallico Parkway	Raleigh
Raleigh Co.	11801 Wayzata Blvd.	Chattanooga

- 2 Drag the column left or right.
- 3 Release the mouse button to drop the column into place.

## Master/Detail One-To-Many forms

Frequently, you have data in one table that is related to data in another table in a one-to-many relationship. For example:

- ◆ You maintain information about departments and their employees. In *one* department there are *many* employees. That is, there is a one-to-many relationship between departments and employees.
- ◆ You maintain information about your customers and their orders. *One* customer typically has *many* orders. There is a one-to-many relationship between customers and orders.



You often want to display this type of relationship in a form. Such a form is called a master/detail one-to-many form.

For example, the following form displays information about one department at the top and all that department's employees at the bottom:

The screenshot shows a software window titled "Departments and Their Employees". At the top, there is a section for "R & D" with input fields for "Department ID: 100" and "Manager ID: 501". A note on the left says "To change departments, click the Next, Prior, First, and Last buttons." Below this is a table of employees. The table has columns for Employee ID, First Name, Last Name, Street, City, State, Zip Code, SSN, and Salary. The Employee ID column is highlighted, and its values are all 100, which is the Department ID.

Employee ID	First Name	Last Name	Street	City	State	Zip Code	SSN	Salary
100	Fran	Whitney	49 East Washington	Needham	MA	02192	017-34-9033	\$45,700
100	Mathew	Cobb	77 Pleasant Street	Waltham	MA	02154	052-34-5739	\$62,000
100	Robert	Breault	58 Cherry Street	Millon	MA	02188	025-48-7623	\$57,490
100	Natasha	Shishov	15 Milk Street	Waltham	MA	02154	043-21-6798	\$72,995
100	Kurt	Driscoll	154 School Street	Waltham	MA	02154	024-60-1768	\$48,023
100	Rodrigo	Guevara	East Main Street	Frammingham	MA	01701	084-32-9990	\$47,998
100	Ram	Cowda	79 Page Street	Nalick	MA	01760	017-34-6122	\$59,840
100	Terry	Melkisetian	87 Oxford Road	Watertown	MA	02172	087-60-2311	\$48,500

### About the ID column

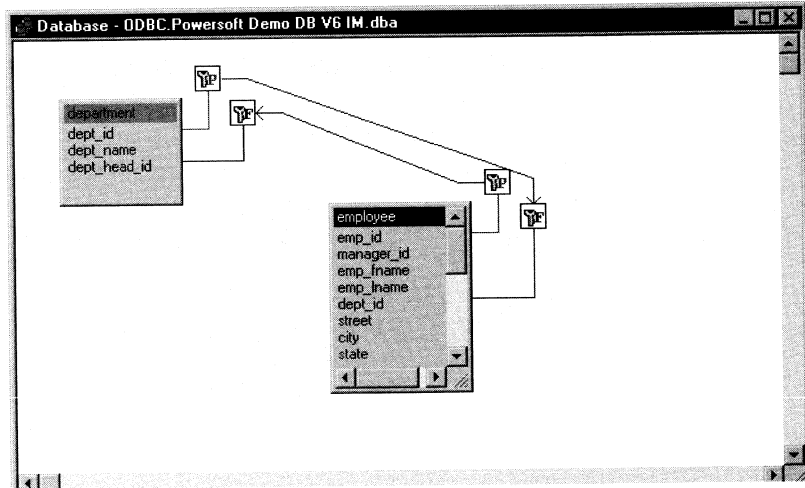
Note the department's ID in the master table at the top of the form.

Although you can't see the department's ID in the detail table at the bottom of the form, it's there. The ID has been hidden by moving grid boundaries to cover the ID column. Since all the ID values are the same, hiding the values is useful.

If the detail table didn't include the ID, the form would not be updatable. If you forget to include necessary columns, InfoMaker prompts you.

The following dialog box shows the database tables that handle the data on this form: the Department table and the Employee table in the Powersoft Demo Database. The Department table is the master table, and the Employee table is the detail table.

Note that there is a primary/foreign key relationship between the tables: the Dept\_id column in the Employee table has the same values as the Dept\_id column in the Department table:



## Master/Detail Many-To-One forms

You might have a lot of information about a particular class of entities, such as customers, employees, or parts. You might want to be able to easily scroll through a list of the entities (the *many*), then see the details for *one* of them. You do that in a master/detail many-to-one form.

Typically, you select one or two columns for the master table (enough for you to identify the entity, such as customer or employee) and the rest of the columns pertaining to the entity for the detail table.

---

### Selecting columns for the master and detail areas

In a master/detail many-to-one form, you usually pick one or two columns for the master area and many columns for the detail area and you change data or insert new data in the detail area only. The data in the master area is usually updated with a different form. To be able to insert new rows in the master area or detail area, you must include all columns that have been defined in the database as requiring values.

FOR INFO For information about defining data so a form can update a database, see "Defining data so a form can update a database" on page 542.

---

For example, the following form lists all customers at the top (the master area) and the details for the selected customer at the bottom (the detail area):

The screenshot shows a window titled "Customer Details" with a table of customer records and a form for the details of the selected customer.

Company Name	Customer ID
The Power Group	101
AMF Corp.	102
Darling Associates	103
P.S.C.	104

Below the table, the details for the selected customer (The Power Group, ID 101) are shown in a form:

Customer: The Power Group 101  
 Address: 3114 Pioneer Avenue  
 City: Rutherford  
 State: NJ  
 Zip Code: 07070  
 Phone Number: (201) 555-8986  
 Contact: Michaels Devlin

A "Print" button is located at the bottom right of the form.

Both the master and detail area use the Customer table in the Powersoft Demo Database. The Company\_name and Id columns were chosen for the master table, and all columns were chosen for the detail table.

#### About the Id column

Since the Id column is the primary key, you must select the Id column for the detail part for the form to be updatable, but you can delete it from the detail part.

## Creating and saving forms

The first step in designing a form is to create a basic form. The procedure is similar for most forms whether you're using a built-in style or a custom form style developed in your organization using PowerBuilder.

### Creating basic forms

A basic form is like a draft that you refine until you have exactly what you want. After you create the basic form, you enhance the form to make data entry fast and to present the data usefully.

❖ **To create a basic form:**

- 1 Click the Form button in the PowerBar.

The Select Form dialog box displays. It lists the forms you have created and has a New button for creating new forms.

- 2 Click New.

The New Form dialog box displays. It shows the data sources and form styles you can choose.

- 3 Choose the data source for the form:

<b>Data source</b>	<b>Pick this data source when</b>
Quick Select	The data is from tables that are connected through a key, and you only need to sort and limit data
SQL Select	Your data is from tables that are not connected through a key, or you want more control over the SQL SELECT statement generated for the data source
Query	The data has been defined as a query

**FOR INFO** For complete information about using each data source and defining the data for the form, see "Defining the data source" on page 155.

- 4 Choose a form style, then click OK.

You can choose one of InfoMaker's built-in form styles or a custom form style developed in your organization using PowerBuilder.

**FOR INFO** For information on design considerations in developing forms for multiple platforms, see Chapter 20, "Working in a Multiplatform Environment".

- 5 Define the data, then click OK.

---

**If you are told the form is not updatable**

After defining the data for the form, you might see a message box telling you that the form is not updatable.

FOR INFO For information about these situations, see "Defining data so a form can update a database" on page 542.

---

InfoMaker generates the basic form and displays it in the Form painter workspace. Here's the basic form for the freeform form using all columns in the Customer table:

The screenshot shows a window titled "Form - (Untitled)" containing a form with the following fields:

- Customer ID: customer
- First Name: customer\_fname
- Last Name: customer\_lname
- Address: customer\_address
- City: customer\_city
- State: customer\_state
- Zip Code: customer\_zip
- Phone Number: customer\_phone
- Company Name: customer\_company\_name

- 6 Save the form.

FOR INFO For information, see "Saving the form" on page 545.

- 7 Run the form.

FOR INFO For information, see "Running forms" on page 546.

FOR INFO At this point, you can enhance the form. For more information, see Chapter 16, "Enhancing Forms".

## Creating a master/detail form

The following steps and screens show how to create a master/detail one-to-many or master/detail many-to-one form.

❖ **To create a master/detail one-to-many or master/detail many-to-one form:**

- 1 Click the Form button in the PowerBar.  
The Select Form dialog box displays.
- 2 Click the New button.  
The New Form dialog box displays.
- 3 Select Quick Select and either of the master/detail styles, then click OK.

---

**You must use Quick Select**

You must use Quick Select and you can select only one master table and one detail table when creating the form. After the master/detail form is created, if you want to add data from another table, you can modify the data source and add new columns.

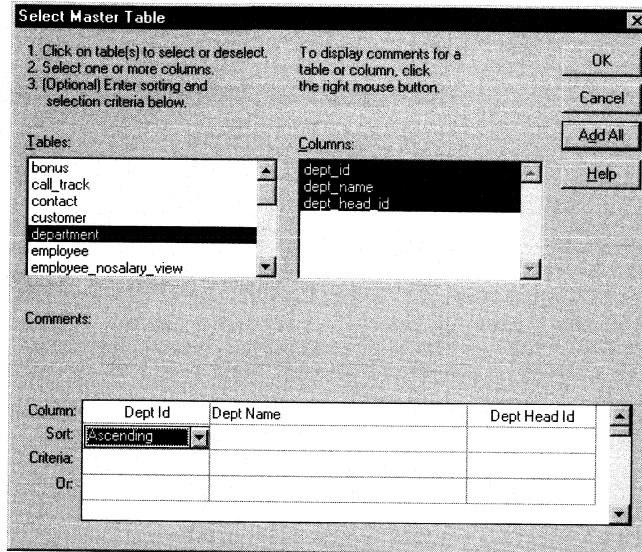
FOR INFO For information about modifying the data source, see Chapter 16, "Enhancing Forms".

---

The Select Master Table dialog box displays.

- 4 Select the master table.  
For a master/detail one-to-many form, this is the table whose data displays one row at a time at the top of the form. For a master/detail many-to-one form, this is the table whose data displays in a list at the top of the form. In the example, it is the Department table.

- 5 Select some or all of the columns in the master table, then click OK:



### If you are told the form is not updatable

After selecting the columns, you might see a message box telling you that the data source is not updatable.

**FOR INFO** For information about these situations, see "Defining data so a form can update a database" on page 542.

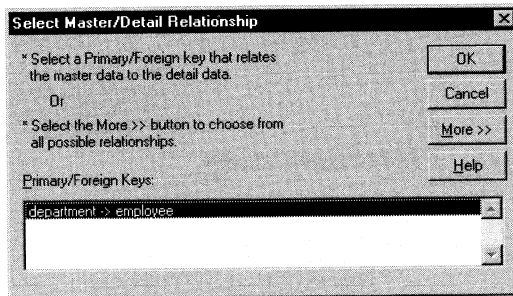
The Select Detail Table dialog box displays.

- 6 Select the detail table.

This is the table that is related to the master table and whose data displays at the bottom of the form. For a master/detail one-to-many form, the data displays many rows at a time. For a master/detail many-to-one form, the data displays one row at a time. In the example, it is the Employee table.

- 7 Select some or all of the columns in the detail table, then click OK.

The Select Master/Detail Relationship dialog box displays. It lists primary/foreign key relationships between the master and detail tables:



- 8 Specify the relationship between the master and detail tables.

To do this, you identify which column or columns in the detail table have the same values as the column or columns in the master table.

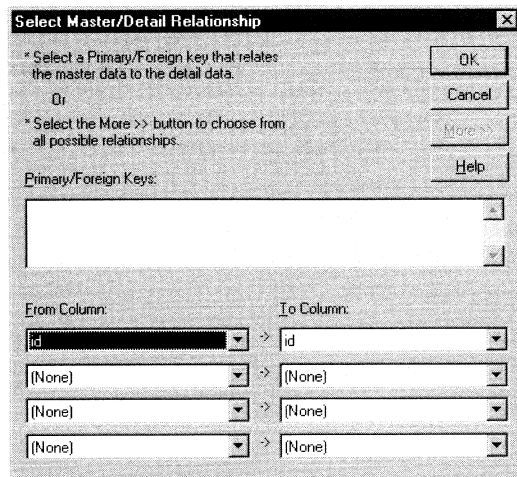
InfoMaker needs this information so it knows which detail rows to display when you display a row in the master table.

- ◆ If there is a primary/foreign key relationship, select it and click OK.

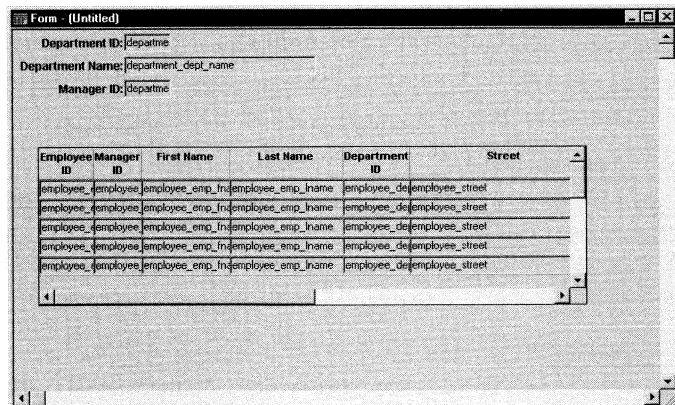


- ◆ If there is no primary/foreign key relationship, click More to specify the relationship. Select one or more columns from the master table and the column or columns in the detail table that contain matching values, and click OK.

When the master table and the detail table are the same table, the complete Select Master/Detail Relationship dialog box displays automatically and you specify the relationship:

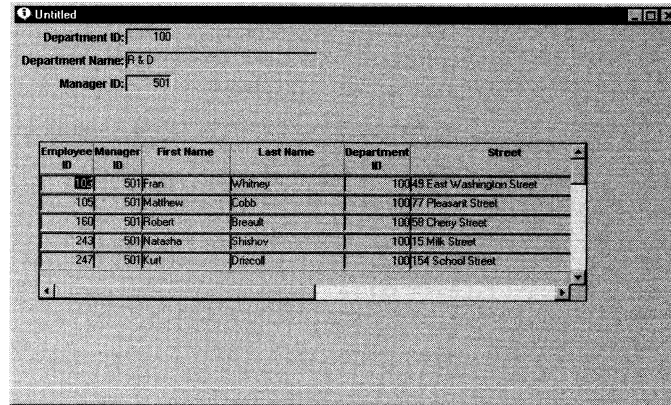


InfoMaker generates the basic form and displays it in the Form painter workspace. The following illustration shows a master/detail one-to-many form:



- 9 Size the master area if necessary and click the Run button to run the form.

When you run the form, the form displays with real data:



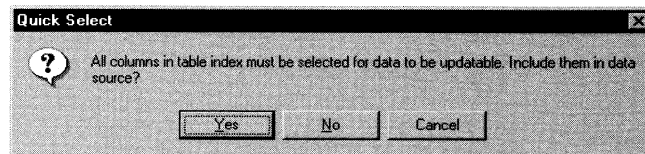
At this point, you can enhance the form. To do so, you first click the Design button to return to the Form painter workspace.

**FOR INFO** For information about how to enhance the form, see Chapter 16, "Enhancing Forms".

## Defining data so a form can update a database

If you want to be able to use a form to update data in a database, you must include all columns that make up a table's unique key when you define the data for the form. This is how InfoMaker identifies rows in the database.

For example, if you are using Quick Select and have not selected all columns in a table's unique key, you see the following dialog box:



You can have InfoMaker add the needed columns automatically by clicking Yes. If you click No and proceed with the ones you originally selected, you will not be able to update data in the database unless you modify the data source after you generate the form.

If you are using SQL Select and do not select all the key columns, you are warned, but you can't add the columns automatically; you can edit the data source after the basic form has been generated.

---

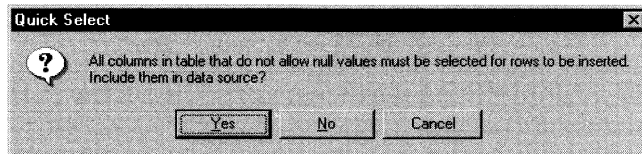
### About the master/detail form styles

The master/detail one-to-many and master/detail many-to-one form styles each have two sources of data, one for the master area and one for the detail area. The data for both the master area and the detail area can be updatable.

---

If you want to be able to insert new rows in a form, you must include all columns that have been defined in the database as requiring values.

For example, if you are using Quick Select and have not selected all columns that allow null values, InfoMaker displays a message box:



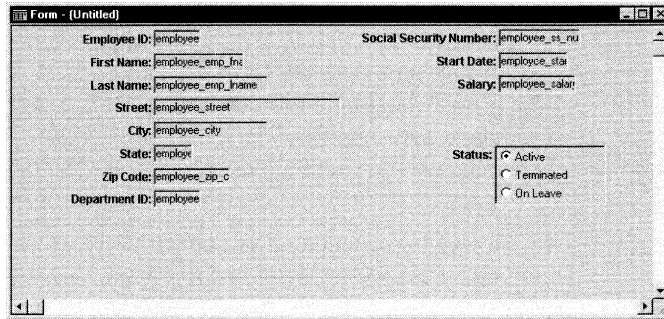
If you want to be able to insert new rows in the form, you can have InfoMaker add the required rows by clicking Yes. You receive this warning only when using the Quick Select data source.

**FOR INFO** For information about controlling updates, see "Controlling updates" on page 555.

## Generating and saving forms

When you finish supplying information about the form style and data source, InfoMaker generates the form and takes you to the Form painter workspace.

Here is the workspace for a freeform form that uses 12 columns of data from the Employee table:



These 12 columns were selected because in the database these columns were defined as requiring values. By selecting these columns, you can use the form to insert new data in the Employee table.

FOR INFO For information about updating data in a form, see "Defining data so a form can update a database" on page 542.

When generating the basic form, InfoMaker uses the information from the Powersoft repository.

## About the Powersoft repository and forms

The Powersoft repository is a set of system tables maintained by the Database painter in InfoMaker or PowerBuilder. The repository contains information about database tables and columns. Repository information extends database definitions.

When creating a form, InfoMaker uses the following information from the Powersoft repository:

For	InfoMaker uses
Tables	Fonts specified for labels, headings, and data
Columns	Text specified for labels and headings Display formats Edit styles

For example, labels and headings you defined for columns in the Database painter are used in the generated form. Similarly, if you associated an edit style with a column in the Database painter, that edit style is automatically used for the column in the form.

**FOR INFO** For more information about the repository, see Chapter 3, "Managing the Database" and the Appendix, "The Powersoft Repository".

At this point, you have a functioning form. You should save it before making any changes.

## Saving the form

When you have generated a form and are in the Form painter workspace, you should save the form. The first time you save it you will give it a name. As you work, you should save your form frequently so that you don't lose changes.

### ❖ To save the form:

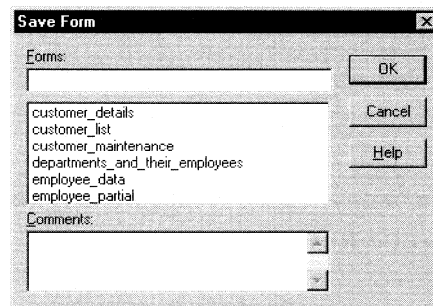
- 1 Click the Save button.

*or*

Select File>Save from the menu bar.

If you have previously saved the form, InfoMaker saves the new version and returns you to the Form painter workspace.

If you have not previously saved the form, InfoMaker displays the Save Form dialog box:



- 2 Name the form in the Forms box.

The form name can be any valid identifier up to 40 characters.

**FOR INFO** For information about InfoMaker identifiers, see Chapter 21, "Identifiers".

- 3 Enter comments to describe the form.
- 4 Click OK.

InfoMaker saves the form in the current library.

## Working with forms

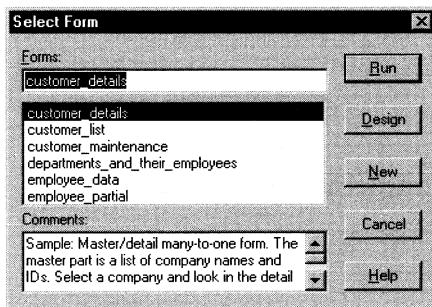
Once you've created a basic form, you can run it, import data into it, save its data in an external file, and print it. You can also assign actions to buttons you place in the form.

## Running forms

You run forms to display and change information in the database. You can run a form anytime after InfoMaker has generated a basic form.

❖ **To run a form you are not currently working on:**

- ◆ Click the Open button in the PainterBar, select the form in the Select Form dialog box, then click Run:



❖ **To run a form you are currently working on:**

- ◆ Click the Run button in the PainterBar.  
*or*  
Select Design>Run from the menu bar.  
*or*  
Press CTRL+R.

---

### On Macintosh

On the Macintosh, use the COMMAND key instead of the CTRL key.

---

What happens

You are now running the form. Command buttons and picture buttons you have placed in the form are now active.

FOR INFO For information about adding objects to a form, see Chapter 16, "Enhancing Forms".

Exactly what you can do when you run the form depends on the form style. It is the form style that determines the menu items that display in the menu bar and the buttons that display in the PainterBar when you run a form.

Typically when you run a form, InfoMaker retrieves the data from the database and displays it to you; then you can scroll through existing data and add, modify, and delete rows in the database.

---

**When you change data using a form**

If you change data and you don't click the Update button before closing the form, InfoMaker prompts you to save data in the database before closing the form.

---

**FOR INFO** For complete information about what you can do when running forms that use InfoMaker's built-in form styles, see "Actions in forms" on page 550.

If you need information about running forms using a style developed by a PowerBuilder user at your site, see the developer.

## Limiting the retrieved data

After you run a form, you can specify criteria that limit the rows of data and cause a re-retrieval of rows.

❖ **To specify criteria to limit the data:**

1 Click the Criteria button in the toolbar.

*or*

Select Rows>Specify Criteria from the menu bar.

InfoMaker clears all the data.

2 Specify the criteria.

In master/detail forms, you specify the criteria in the form's master area, which is a freeform area for master/detail one-to-many forms and a grid area for master/detail many-to-one forms. Both are like the grid you use when defining data using the Quick Select data source.

Use expressions and operators in the blank spaces or the grid to specify criteria.

To specify a second set of criteria in a master/detail one-to-many form, press PAGE DOWN to get a new entry form and specify the criteria in the new master area. The second set of criteria will be ORed with the first set; data is retrieved if one set *or* the other set is true.

FOR INFO For information about expressions, see "Using Quick Select" on page 156, and Chapter 22, "Operators and Expressions".

3 Click the Apply button.

*or*

Select Rows>Apply Criteria from the menu bar.

InfoMaker retrieves rows based on the criteria.

4 If you want to modify the criteria, repeat steps 1 through 3.

## Importing data into a form

When you run a form, you can import data from a file and save it in the database.

---

### For freeform and grid forms only

You can import data only in freeform and grid forms. You cannot import data in master/detail forms.

---

❖ **To import data:**

1 Select Rows>Import from the menu bar.

The Select Import File dialog box displays.

2 Navigate to the folder you want and select the file from which you want to import the data.

The types of files that you can import into the form are shown in the Files of Type dropdown listbox.



3 Click Open.

InfoMaker reads the data from the file. You can view the data and save it in an external file.

---

**Data from a file must match retrieved columns**

When importing data from a file, the data must match all the columns in the retrieved data (the columns specified in the SELECT statement), not just the columns that are displayed in the form.

---

## Saving data in an external file

When you run a form, you can save the data retrieved (and optionally the headers) in an external file.

❖ **To save data in an external file:**

- 1 Select File>Save Rows As from the menu bar.

The Save As dialog box displays.

- 2 Choose a format for the file from the Save As Type dropdown listbox.

When you choose a format, InfoMaker supplies the appropriate file extension.

If you want the column headers saved in the file, select a file format that includes headers, for example Excel With Headers. When you select a *with headers* format, the names of the database columns (not the column labels), will also be saved in the file.

---

**Saving the data as a Powersoft report or HTML table**

Choose Powersoft Report in the Save As Type box to save the data as a Powersoft PSR file. Choose HTML Table in the Save As Type box to save the data as an HTM file.

**FOR INFO** For information about PSR files and about working with files saved as HTM files, see Chapter 5, "Enhancing Reports".

---

- 3 Name the file.

- 4 Click Save.

InfoMaker saves all rows in the file; all columns in the rows are saved.

## Printing forms

Although forms are primarily used for data entry, after you run a form you can print the form. Printing a freeform form is particularly helpful because each page displays the data one row at a time.

### ❖ To print a form:

- ◆ Select File>Print from the menu bar.

*or*

Add a command button to a form, associate the print action with the button, and click the button.

**FOR INFO** For information about adding a command button and associating an action with a button, see Chapter 16, "Enhancing Forms".

## Actions in forms

You can assign the actions in the following table to buttons you place in a form. Many actions are also available in the toolbar and menu when you run a form.

Because all data is visible in Grid style forms, there are no First, Last, Next, and Prior buttons/actions as there are in the Freeform, Master/Detail One-To-Many, and Master/Detail Many-To-One styles.

**FOR INFO** For information about placing buttons in a form, see Chapter 16, "Enhancing Forms".

Action	Menu item	What the action does	Form styles available in
Apply_Criteria	Rows>Apply Criteria	Validates the selection criteria, then re-retrieves the rows based on the criteria	All
Cancel_Updates	Rows>Cancel Changes	Discards changes made since the last update	All
Clear_Filter	None	Clears the current filter	Freeform, Grid
Clear_Detail_Filter	None	Clears the detail filter	Master/Detail One-To-Many, Master/Detail Many-To-One

<b>Action</b>	<b>Menu item</b>	<b>What the action does</b>	<b>Form styles available in</b>
Clear_Master_Filter	None	Clears the master filter	Master/Detail One-To-Many, Master/Detail Many-To-One
Close	File>Close	Closes the form (in the InfoMaker environment, returns you to design mode)	All
Delete_Row	Rows>Delete (CTRL+D)	Deletes the current row in the form	All
Filter_Dialog	None	Displays the Filter dialog box for defining a filter	Freeform, Grid
Filter_Detail_Dialog	None	Displays the Filter dialog box for defining a filter for the detail part of the form	Master/Detail One-To-Many, Master/Detail Many-To-One
Filter_Master_Dialog	None	Displays the Filter dialog box for defining a filter for the master part of the form	Master/Detail One-To-Many, Master/Detail Many-To-One
First_Row	Rows>First	Scrolls to the first retrieved row (in the master area)	Freeform, Master/Detail One-To-Many
Import_File	Rows>Import	Displays the Select Import File dialog box to select a file for importing rows of data	Freeform, Grid
Insert_Row	Rows>Insert (CTRL+I)	Inserts a new row (in the master area) and scrolls to it with the cursor in the first column	All
Last_Row	Rows>Last	Scrolls to the last row in the form (in the master area)	Freeform, Master/Detail One-To-Many
Next_Row	Rows>Next	Scrolls to the next row	Freeform, Master/Detail One-To-Many
Print	File>Print	Prints the retrieved data	All

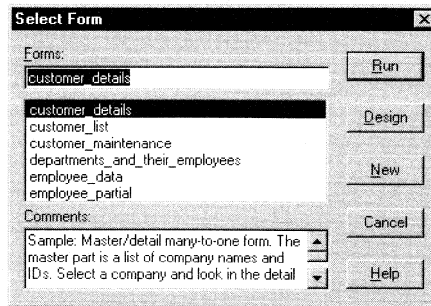
Action	Menu item	What the action does	Form styles available in
Print_Setup	File>Print Setup	Opens the Windows Printer Setup dialog box allowing you to change the printer or its settings	All
Prior_Row	Rows>Get Prior	Scrolls to the previous row in the master area	Freeform, Master/Detail One-To-Many
Retrieve	Rows>Retrieve	Retrieves rows from the database and prompts you for criteria if Prompt for Criteria was specified in the workspace	All
Save_As	File>Save Rows As	Saves rows of data in a selected file format, including text format or a Powersoft report	Freeform, Grid
Sort_Dialog	None	Displays the Sort dialog box for defining sorting	Freeform, Grid
Sort_Detail_Dialog	None	Displays the Sort dialog box for defining sorting for the detail part of the form	Master/Detail One-To-Many, Master/Detail Many-To-One
Sort_Master_Dialog	None	Displays the Sort dialog box for defining sorting for the master part of the form	Master/Detail One-To-Many, Master/Detail Many-To-One
Specify_Criteria	Rows>Specify Criteria	Updates the database, then clears the form to allow you to specify selection criteria for the rows  FOR INFO For information, see "Limiting the retrieved data" on page 547	All
Update_Row	Rows>Update (CTRL+I)	Updates the current row in the database	Freeform
Update_Rows	Rows>Update (CTRL+U)	Updates all modified rows in the database	Grid, Master/Detail One-To-Many, Master/Detail Many-To-One

## Accessing and deleting existing forms

### ❖ To access an existing form:

- 1 Click the Open button in the PowerBar.

The Select Form dialog box displays. It lists the forms you have created and has Run and Design buttons for accessing existing forms:



- 2 Highlight the form you want in the list.
- 3 Do one of the following:
  - ◆ Click the Design button to display the form in the workspace. You can work on the design of the form.  
FOR INFO For more information, see Chapter 16, "Enhancing Forms".
  - ◆ Click the Run button to run the form. When you run a form, InfoMaker retrieves data from the database and displays the form so you can view and change information in the database.  
FOR INFO For more information, see "Running forms" on page 546.

### ❖ To delete an existing form:

- 1 In the workspace, select File>Delete from the menu bar.  
The Delete Form dialog box displays.
- 2 Select the form you want to delete.
- 3 Click OK.

---

**Deleting multiple forms**

If you want to delete multiple forms, it's easiest to use the Environment painter.

FOR INFO For information about the Environment painter, see Chapter 2, "Managing Your Environment".

---

## Controlling updates

When InfoMaker generates the basic form, it defines whether the data is updatable by default as follows:

- ◆ If the form contains columns from a single table and includes that table's key columns, InfoMaker defines all columns as updatable and specifies a nonzero tab order for each column, allowing you to tab to the columns.
- ◆ If the form contains columns from two or more tables or from a view, InfoMaker defines all columns as not being updatable and sets all tab orders to zero, preventing you from tabbing to them.

You can accept the default settings or modify the update characteristics for a form.

---

### **Data sources in the master/detail styles**

The master/detail styles have a data source for the master area and a data source for the detail area. Each of the data sources can be updatable.

---

## What you can do

You can:

- ◆ Allow updates in a form associated with multiple tables or a view; you can define one of the tables as being updatable
- ◆ Prevent updates in a form associated with one table
- ◆ Prevent updates to specific columns in a form that is associated with an updatable table
- ◆ Specify which columns uniquely identify a row to be updated
- ◆ Specify which columns will be included in the WHERE clause of the UPDATE or DELETE statement InfoMaker generates to update the database
- ◆ Specify whether InfoMaker generates an UPDATE statement, or a DELETE then an INSERT statement, to update the database when you modify the values in a key column

---

### Updatability of views

Some views are logically updatable; some are not.

FOR INFO For the rules your DBMS follows for updating views, see your DBMS documentation.

---

#### ❖ To specify update characteristics for a form:

- 1 Select Rows>Update Properties from the menu bar.  
The Specify Update Properties dialog box displays.
- 2 To prevent updates to the data, make sure the Allow Updates box is not selected.  
To allow updates, select the Allow Updates box and specify the other settings as described below.
- 3 Click OK.  
InfoMaker returns you to the workspace.


#### Changing tab values

InfoMaker does not change the tab values associated with columns after you change the update characteristics of the form. So if you have allowed updates to a table in a multitable form, you should change the tab values for the updatable columns so you can tab to them.

FOR INFO For more information, see "Defining the tab order in a form" on page 587.

## Specifying the table to update

Each form can update one table:



The image shows a dialog box with the label 'Table to Update:' followed by a text field containing the word 'employee' and a small square button with a downward-pointing arrow.

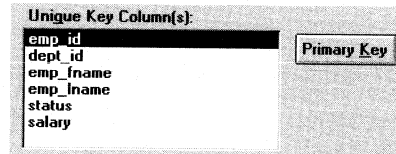
#### ❖ To specify the table that can be updated using a form:

- 1 Select Rows>Update Properties from the menu bar.  
The Specify Update Properties dialog box displays.
- 2 Select the table from the Table to Update box.



## Specifying the unique key columns

The Unique Key Columns box in the Specify Update Properties dialog box specifies which columns InfoMaker uses to identify a row being updated. InfoMaker uses the column or columns you specify here as the key columns when generating the WHERE clause to update the database (as described below):



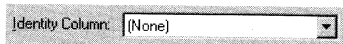
The key columns you select here must uniquely identify a row in the table. They can be the table's primary key, though they don't have to be.

### Using the primary key

Clicking the Primary Key button cancels any changes in the Unique Key Columns box and highlights the primary key for the updatable table.

## Specifying an identity column

Many DBMSs allow you to specify that the value for a column in a new row is to be automatically assigned by the DBMS. This kind of column is called an **identity column**. Different DBMSs provide different types of identity columns. For example, some DBMSs allow you to define autoincrement columns so that the column for a new row is automatically assigned a value one greater than that of the previous highest value. You could use this feature to specify that order number be automatically incremented when someone adds a new order:

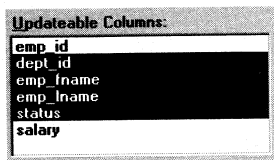


By specifying an identity column in the Specify Update Properties dialog box, you tell InfoMaker to bring back the value of a new row's identity column after an insert in the form so you can see it.

**FOR INFO** For information about identity columns in your DBMS, see your DBMS documentation.

## Specifying updatable columns

You can make all or some of the columns in a table updatable:



Updatable columns are displayed highlighted. In the preceding screen, Dept\_ID, Emp\_Fname, Emp\_Lname, and Status are updatable. The other columns are not.

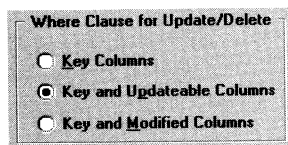
Click a nonupdatable column to make it updatable. Click an updatable column to make it nonupdatable.

### Changing tab values

If you have changed the updatability of a column, you should change its tab value in the workspace. For example, if you have allowed a column to be updated, you should change its tab value to a nonzero number so you can tab to it.

## Specifying the WHERE clause for update/delete

Sometimes multiple users are accessing the same tables at the same time. In these situations, you need to decide when to allow your form to update the database. If you allow your form to always update the database, it could overwrite changes made by other users:



You can control when updates succeed by specifying which columns InfoMaker includes in the WHERE clause in the UPDATE or DELETE statement used to update the database:

```
UPDATE table...  
SET column = newvalue  
WHERE col1 value1  
AND col2 = value2 ..
```

```
DELETE
FROM table
WHERE col1 = value1
AND col2 = value2 ...
```

---

### Using timestamps

Some DBMSs maintain timestamps so you can ensure that you are working with the most current data. If the SELECT statement for the form contains a timestamp column, InfoMaker includes the key column and the timestamp column in the WHERE clause for an UPDATE or DELETE statement regardless of which columns you specify in the Where Clause for Update/Delete box.

If the value in the timestamp column changes (possibly due to another user modifying the row), the update fails.

FOR INFO To see whether you can use timestamps with your DBMS, see *Connecting to Your Database*.

---

Choose one of the following in the Where Clause for Update/Delete box (the results are illustrated by an example following the table):

Option	Result
Key Columns	<p data-bbox="602 894 1224 953">The WHERE clause includes the key columns only (these are the columns you specified in the Unique Key Columns box)</p> <p data-bbox="602 963 1247 1072">The values in the originally retrieved key columns for the row are compared against the key columns in the database. No other comparisons are done. If the key values match, the update succeeds</p> <hr/> <p data-bbox="602 1117 686 1140"><b>Caution</b></p> <p data-bbox="602 1151 1237 1287">Be very careful when using this option. If you tell InfoMaker only to include the key columns in the WHERE clause, if someone else modified the same row after you retrieved it, their changes will be overwritten when you update the database (see the example following this table).</p> <hr/> <p data-bbox="602 1332 1214 1414">Use this option only with a single-user database or if you are using database locking. In other situations, choose one of the other two options described in this table</p>

Option	Result
Key and Updateable Columns	The WHERE clause includes all key and updatable columns The values in the originally retrieved key columns and the originally retrieved updatable columns are compared against the values in the database. If any of the columns have changed in the database since the row was retrieved, the update fails
Key and Modified Columns	The WHERE clause includes all key and modified columns The values in the originally retrieved key columns and the modified columns are compared against the values in the database. If any of the columns have changed in the database since the row was retrieved, the update fails

### Example

Consider this situation: a form is updating the Employee table, whose key is Emp\_ID; all columns in the table are updatable. Say you have changed the salary of employee 1001 from \$50,000 to \$65,000. This is what happens with the different settings for the WHERE clause columns:

- ◆ If you choose Key Columns for the WHERE clause, the UPDATE statement looks like this:

```
UPDATE Employee
SET Salary = 65000
WHERE Emp_ID = 1001
```

This statement will succeed *regardless of whether other users have modified the row since your form retrieved the row*. For example, if another user had modified the salary to \$70,000, that change will be overwritten when your form updates the database.

- ◆ If you choose Key and Modified Columns for the WHERE clause, the UPDATE statement looks like this:

```
UPDATE Employee
SET Salary = 65000
WHERE Emp_ID = 1001
AND Salary = 50000
```

Here the UPDATE statement is also checking the original value of the modified column in the WHERE clause. The statement will fail if another user changed the salary of employee 1001 since your form retrieved the row.

- ◆ If you choose Key and Updateable Columns for the WHERE clause, the UPDATE statement looks like this:

```
UPDATE Employee
SET Salary = 65000
```

```

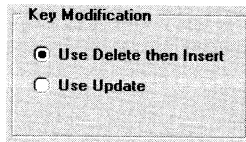
WHERE Emp_ID = 1001
      AND Salary = 50000
      AND Emp_Fname = original_value
      AND Emp_Lname = original_value
      AND Status = original_value
      ...

```

Here the UPDATE statement is checking all updatable columns in the WHERE clause. This statement will fail if any of the updatable columns for employee 1001 have been changed since your form retrieved the row.

## Specifying update when key is modified

The Key Modification property determines the SQL statements InfoMaker generates whenever a key column—a column you specified in the Unique Key Columns box—is changed:



The options are:

- ◆ Use DELETE then INSERT (default)
- ◆ Use UPDATE

How to choose a setting

Consider the following when choosing the Key Modification setting:

- ◆ If multiple rows are changed, DELETE and INSERT always work. In some DBMSs, UPDATE will fail if the user modifies two keys and sets the value in one row to the original value of the other row.
- ◆ You might choose the setting here based on your DBMS triggers. For example, if there is an Insert trigger, you should select Use Delete then Insert.
- ◆ If only one row can be modified by the user before the database is updated, use UPDATE; it is faster.



About this chapter

Before using a form, you might want to enhance it to make it easier to use and interpret data. This chapter describes enhancements you can make to a form.

Contents

<b>Topic</b>	<b>Page</b>
About enhancing forms	564
Working in the Form painter workspace	565
Reorganizing objects in the form	574
Modifying general form properties	582
Adding objects to the form	594
Highlighting information in a form	608
Displaying and validating data in a form	610

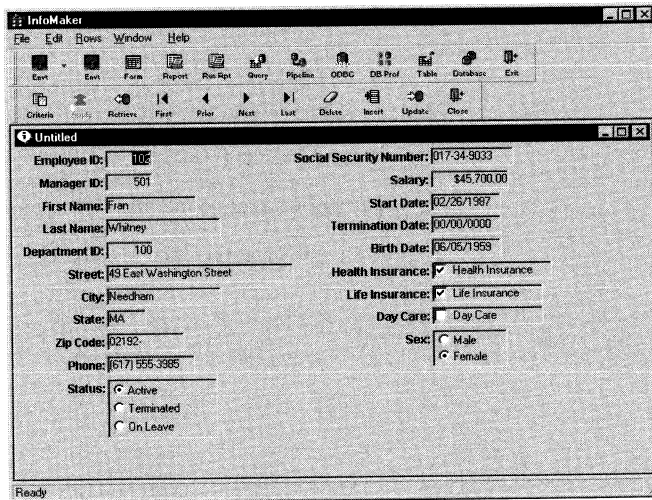
## About enhancing forms

InfoMaker provides you with a variety of ways to enhance a form to make it easier to enter data and to interpret data. You can customize the display of the data. You can also place objects such as command buttons and pictures in a form.

You enhance a form in the workspace and then you run the form to see if it's usable and what it looks like with data. You can repeat this process many times until you have exactly what you want.

❖ **To enhance a form:**

- 1 Create the basic form and click the Run button to see the form with data:

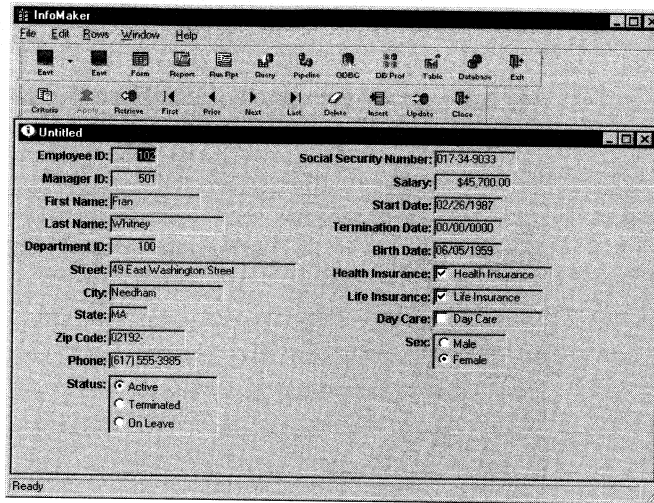


- 2 Click the Design button to return to the Form painter workspace, where you can enhance the form.



## Working in the Form painter workspace

Here is the Form painter workspace for a Freeform form style:



This section describes how to work in the Form painter workspace:

- ◆ "Using the Form painter toolbars" next
- ◆ "Using the popup menus in the Form painter" on page 567
- ◆ "Using property sheets in the Form painter" on page 568
- ◆ "Using keyboard shortcuts in the Form painter workspace" on page 568
- ◆ "Selecting objects in the Form painter" on page 570
- ◆ "Defining default colors and borders in the Form painter" on page 571
- ◆ "Printing the form definition" on page 573

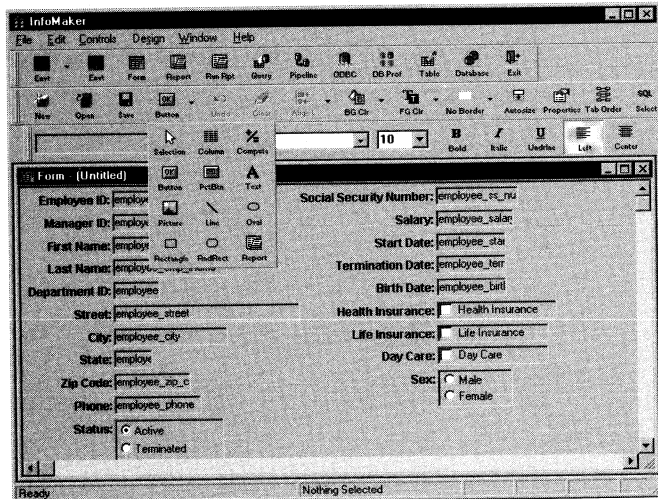
## Using the Form painter toolbars

The Form painter has a customizable PainterBar and a StyleBar. The PainterBar has buttons for form operations. The StyleBar has buttons for modifying text properties such as the font and whether the text is bold or italic.

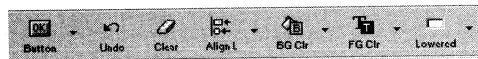
**FOR INFO** For information on using toolbars, and customizing and manipulating them, see "Using toolbars" on page 24.

## About the PainterBar

The Form painter's PainterBar has buttons for operations such as New, Open, Close, and Run. It also has five dropdown toolbars, which have a small black triangle on the right side of the button. When you click the triangle, the dropdown toolbar displays and you can click a button in the toolbar:



Here's part of the PainterBar that shows the five dropdown toolbars:



**Controls dropdown toolbar** has the controls you can add to a form.

**Layout dropdown toolbar** has alignment, sizing, and spacing options you can choose for selected objects.

**Background Color dropdown toolbar** has background colors that you can choose for one or more selected objects.

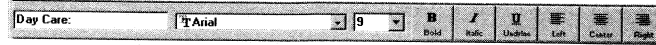
**Foreground Color dropdown toolbar** has foreground colors that you can choose for one or more selected objects. In a text object, the foreground color specifies the color of the text.

**Borders dropdown toolbar** has borders you can choose for one or more selected objects.

For example, to place a computed field in the form, click the Compute button on the Controls dropdown toolbar, then click the location in the form where you want the computed field to appear.

## About the StyleBar

The Form painter also has a toolbar called the StyleBar, which you can use to modify text properties in the form:



Like other toolbars, you can move and place the StyleBar where it's convenient for you.

## Using the popup menus in the Form painter

Each element of the form (such as text, columns, computed fields, buttons, even the form itself) has a popup menu you can use to perform appropriate actions and access the associated property sheet.

### ❖ To use a popup menu in the Form painter workspace:

- 1 Position the pointer over the object or the background of the form.
- 2 Press the right mouse button.

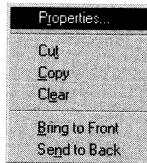
---

#### On Macintosh

On the Macintosh, press the control button while you click the mouse.

---

Here is the popup menu for a column of data or a text object in a freeform form:



- 3 Click the menu item you want.

To access the object's property sheet, you select Properties from the popup menu. To perform an action, for example Cut, you select the action.

## Using property sheets in the Form painter

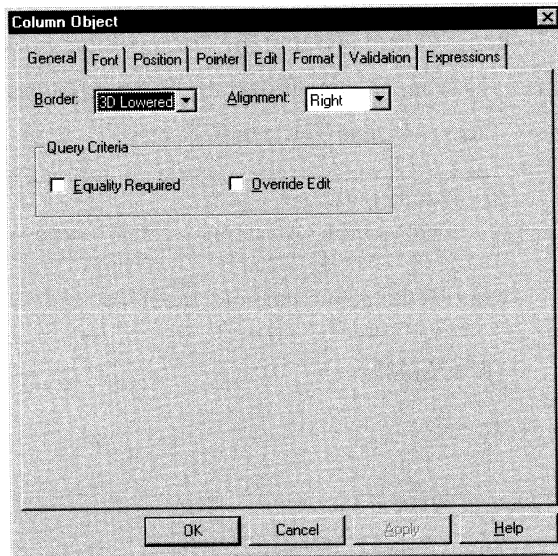
All objects in a form (such as text, columns, computed fields, buttons, and so on) and the workspace itself have a property sheet you can use to modify the form. The items in the property sheet are appropriate to the object you are working on.

❖ **To use a property sheet in the Form painter:**

- 1 Position the pointer over the part you want to modify.
- 2 Display the popup menu and select Properties.

The appropriate property sheet displays.

Here's the property sheet for a column. It has eight tabbed property pages of information, which you access by clicking one of the tabs. For example, to choose an edit style for a column, you click the Edit tab:



When you want to modify part of a form, display the property sheet and pick the item you want to change. Click the various tabs to change pages.

## Using keyboard shortcuts in the Form painter workspace

The following table lists the keyboard shortcuts available in the Form painter workspace:

<b>Action</b>	<b>Windows shortcut</b>	<b>Macintosh shortcut</b>	<b>Comments</b>
Boldface text	CTRL+B		Toggles boldface on and off
Center text	CTRL+SHIFT+C		
Clear	DEL	DEL	Deletes all selected objects
Close painter	CTRL+F4		
Close InfoMaker	ALT+F4	COMMAND +Q	Available everywhere in InfoMaker
Copy selected objects	CTRL+C	COMMAND +C	
Cut selected objects	CTRL+X	COMMAND +X	
Edit text	CTRL+E		Activates the Text box in the StyleBar
File editor	SHIFT+F6	SHIFT+F6	Available everywhere in InfoMaker
Font family	CTRL+F		Activates the Font box in the StyleBar
Italicize text	CTRL+I		Toggles italics on and off
Left justify text	CTRL+SHIFT+L		
Move object	ARROW	ARROW	
Next child window	CTRL+F6		
Paste contents of the clipboard	CTRL+V	COMMAND +V	A cut or copy action placed objects on the clipboard
PowerPanel	CTRL+SHIFT+N		Available everywhere in InfoMaker
Resize object	SHIFT+ARROW	SHIFT+ARROW	
Return focus to control	CTRL+O		Returns focus to the control from the StyleBar
Right justify text	CTRL+SHIFT+R		
Run the form	CTRL+R		

Action	Windows shortcut	Macintosh shortcut	Comments
Select all	CTRL+A	COMMAND +A	Selects all objects in form
Switch to	CTRL+ESC, ALT+ESC		Available everywhere in InfoMaker
Underline text	CTRL+U		Toggles underline on and off
Undo/Redo	CTRL+Z	COMMAND +Z	Undoes the most recent change (including the most recent undo)

## Selecting objects in the Form painter

The Form painter provides several ways for you to select objects to act on. You can select multiple objects and can perform some actions on all the selected objects as a unit. For example, you can move all of them or change the fonts used to display text for all of them.

❖ **To select one object in a form:**

- ◆ Click it.

The object displays with handles on it. Previously selected objects are no longer selected.

❖ **To select neighboring multiple objects in a form:**

- 1 Press and hold the left mouse button (mouse button on the Macintosh) at one corner of the neighboring objects.
- 2 Drag the mouse over the objects you want to select.  
InfoMaker displays a bounding box.
- 3 Release the mouse button.  
All the objects in the region are selected.

❖ **To select non-neighboring multiple objects in a form:**

- 1 Click the first object.

- 2 Press and hold the CTRL key (COMMAND key on the Macintosh) and click additional objects.

All the objects are selected.

❖ **To select all objects in a form:**

- ◆ Select Edit>Select All from the menu bar.  
*or*  
Press CTRL+A (COMMAND+A on the Macintosh).

## Displaying information about selected objects in a form

To help you move and resize objects, InfoMaker displays information about the position of selected objects.

❖ **To display information about the position of an object:**

- ◆ Select the object.

InfoMaker displays the object name, x and y coordinates, width, and height in MicroHelp:

Name: employee_city	x: 673	y: 597	w: 526	h: 61
---------------------	--------	--------	--------	-------

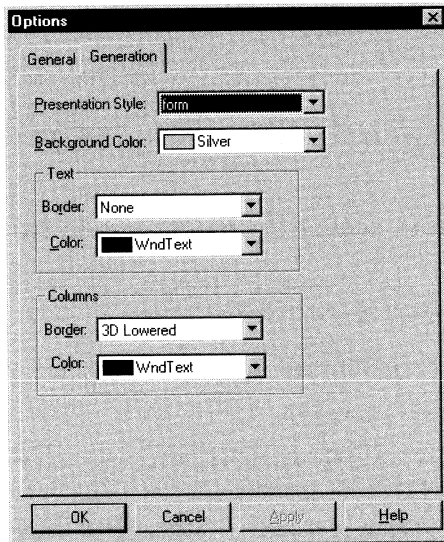
If you select more than one object, InfoMaker displays *Group Selected* in the Name area, and coordinates and size do not display.

## Defining default colors and borders in the Form painter

For Freeform and Grid style forms, you can specify the default colors and borders that InfoMaker uses when generating a basic form in the workspace. Each style can have its own defaults. You can override a style's defaults for a particular form in the workspace.

❖ **To specify default colors and borders for a Freeform or Grid style form:**

- 1 Select Design>Options from the menu bar and then select the Generation page:



- 2 Select the style whose defaults you want to change from the Presentation Style dropdown listbox.
- 3 Change one or more of the following properties:

Property	Meaning
Background color	The default color for the background of the form
Text border and color	The default border and color used for labels and headings in the form
Column border and color	The default border and color used for data values in the draft form

- 4 Click OK.

You return to the workspace.

From now on, when you create a new form using the style, InfoMaker uses the defaults you just specified for colors and borders when generating the basic form.



**FOR INFO** For more information about colors, see "Setting colors for a form" on page 583. For more information about borders, see "Using borders in a form" on page 589.

## Printing the form definition

At any point when you are working on a form, you can print a document that lists all objects in the form and their properties.

- ❖ **To print a document describing the objects in the form:**
  - ◆ Select File>Print Form Definition from the menu bar in the Form painter workspace.  
InfoMaker prints the form definition on the default printer.
- ❖ **To select a different printer:**
  - ◆ Select File>Printer Setup from the menu bar and select the printer.

## Reorganizing objects in the form

This section describes how you can change the layout of any of the objects in a form.

- ◆ "Using the grid in the Form painter" next
- ◆ "Deleting objects in the Form painter" on page 575
- ◆ "Moving objects in the Form painter" on page 576
- ◆ "Copying and pasting objects in the Form painter" on page 576
- ◆ "Resizing objects in the Form painter" on page 577
- ◆ "Aligning objects in the Form painter" on page 578
- ◆ "Equalizing the space between objects in the Form painter" on page 579
- ◆ "Equalizing the size of objects in the Form painter" on page 579
- ◆ "Undoing changes in the Form painter" on page 580
- ◆ "Sliding objects in a form" on page 580

## Using the grid in the Form painter

The Form painter provides a grid to help you align objects.

---

### **About the grid**

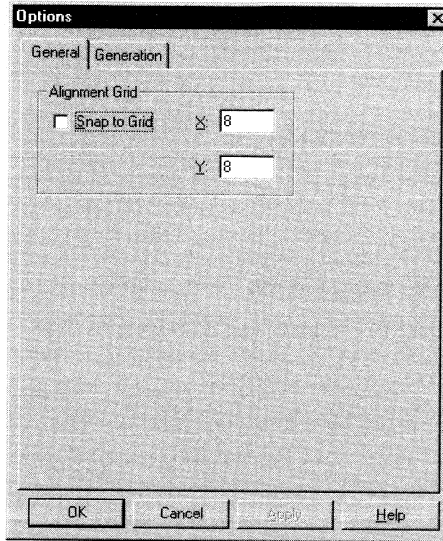
The Form painter grid is invisible. You cannot see it in the workspace, but your objects can snap to the grid as you move them.

---

❖ **To use the grid in the Form painter:**

- 1 Select Design>Options from the menu bar.

The Options property sheet displays:



- 2 Use the options on the General page to:

- ◆ Make objects snap to a grid position when you place them or move them in a form
- ◆ Specify the size (height and width) of the grid cells

The options are:

Option	Meaning
Snap to Grid	If selected, controls snap to the grid when you place or move them
X	The width of each cell in the grid in pixels
Y	The height of each cell in the grid in pixels

## Deleting objects in the Form painter

❖ **To delete objects in the Form painter:**

- 1 Select the objects you want to delete.

- 2 Click the Clear button.  
*or*  
Select Edit>Clear from the menu bar.  
*or*  
Press the DEL key.

## Moving objects in the Form painter

In all form styles  
except Grid

In all form styles except grid, you can move all the objects (such as headings, labels, columns, and drawing objects) anywhere you want.

❖ **To move objects in the Form painter:**

- 1 Select the objects you want to move.
- 2 Do one of the following:
  - ◆ Drag the objects with the mouse.
  - ◆ Press an arrow key to move the objects in one direction.

In grid forms

If you are working in a grid style form, you can reorder columns.

❖ **To reorder columns in the Form painter:**

- 1 If grid lines do not show in the column heading, select Grid Lines from the column heading popup menu.
- 2 Select a column heading.

InfoMaker selects the column and displays a line representing the column border:

Employee ID	Department	Name	Salary	Status
[A shaded horizontal bar with a small icon on the left, representing a column border selection.]				

- 3 Drag the column left or right.
- 4 Release the mouse button to drop the column into position.

## Copying and pasting objects in the Form painter

You can copy objects within a form and to other forms. All properties of the object are copied.

❖ **To copy an object in the Form painter:**

- 1 Select the object in the workspace.
- 2 Select Edit>Copy from the menu bar.  
*or*  
Press CTRL+C (COMMAND +C on the Macintosh).

The object is copied to the clipboard.

- 3 Paste the object.

To paste the object within the same form, select Edit>Paste from the menu bar or press CTRL+V (COMMAND+V on the Macintosh).

To paste the object in another form, open the Form painter again, open the desired form, and paste the object.

InfoMaker pastes the object at the same location as in the source form. If you are pasting into the same form, you should move the pasted object so it doesn't overlay the original object. InfoMaker displays a message box if the object you are pasting is not valid in the destination form.

---

**Cutting objects**

You can also select one or more objects and cut the objects by selecting Edit>Cut from the menu bar.

---

## Resizing objects in the Form painter

You can resize an object using the mouse or the keyboard.

### Using the mouse

To resize an object using the mouse, select it, then grab an edge and drag it with the mouse.

### Using the keyboard

To resize an object using the keyboard, select the object and do the following:

To make the object	Press
Wider	SHIFT+RIGHT ARROW
Narrower	SHIFT+LEFT ARROW
Taller	SHIFT+DOWN ARROW
Shorter	SHIFT+UP ARROW

### In grid forms

You can resize columns in grid forms.

❖ **To resize a column in the Form painter:**

- 1 Position the pointer at a column boundary.  
The pointer changes shape to a two-headed arrow.
- 2 Press and hold the left mouse button (mouse button on the Macintosh) and drag the boundary to resize the column.
- 3 Release the mouse button when the column is the correct width.

## Aligning objects in the Form painter

Often you want to align several objects or make them all the same size. You can use the grid to align the objects or have InfoMaker align them for you.

❖ **To align objects in the Form painter:**

- 1 Select the object whose position you want to use to align the others.  
InfoMaker displays handles around the selected object.
- 2 Extend the selection by pressing and holding the CTRL key (COMMAND KEY on the Macintosh) and clicking the objects you want to align with the first one.  
All the objects have handles on them.

---

**Avoid lasso selection for aligning objects**

Avoid selecting objects by dragging the mouse to put a bounding box around multiple objects. You cannot control which object is used as the basis for aligning the other objects.

---

- 3 Use the Layout dropdown toolbar in the PainterBar.  
*or*  
Select Edit>Align Controls from the menu bar.
- 4 Select the dimension along which you want to align the controls.  
For example, to align the objects along the left side, click the Align L button in the Layout dropdown toolbar or select the first choice in the cascading menu.  
InfoMaker moves all the selected objects to align with the first one.

## Equalizing the space between objects in the Form painter

If you have a series of objects and the spacing is fine between two of them but the spacing is wrong for the rest, you can easily equalize the spacing around all the objects.

### ❖ To equalize the space between objects in the Form painter:

- 1 Select the two objects whose spacing is correct.

To do this click one object, then press CTRL (COMMAND on the Macintosh) and click the second object.

- 2 Select the other objects whose spacing you want to be the same as the first two objects by pressing CTRL (COMMAND on the Macintosh) and clicking.

- 3 Use the Layout dropdown toolbar in the PainterBar.

*or*

Select Edit>Space Controls from the menu bar.

- 4 Select the dimension whose spacing you want to equalize.

For example, to equalize the vertical spacing of the objects, click the Space V button in the Layout dropdown toolbar or select the second choice in the cascading menu.

## Equalizing the size of objects in the Form painter

Say you have several objects in a form and want their sizes to be the same. You can accomplish this manually or by using the Edit menu.

### ❖ To equalize the size of objects in the Form painter:

- 1 Select the object whose size is correct.

- 2 Select the other objects whose size you want to match the first object by pressing CTRL (COMMAND on the Macintosh) and clicking the objects.

- 3 Use the Layout dropdown toolbar in the PainterBar.

*or*

Select Edit>Size Controls from the menu bar.

- 4 Select the dimension whose size you want to equalize.

For example, to equalize the width of the objects, click the Size W button in the Layout dropdown toolbar or select the first choice in the cascading menu.

## Undoing changes in the Form painter

You can undo your most recent change (a move, size, or clear action) in the workspace.

### ❖ To undo a change in the Form painter:

- ◆ Select Edit>Undo Size/Move.

*or*

Edit>Undo Clear from the menu bar.

*or*

Press CTRL+Z (COMMAND+Z on the Macintosh).

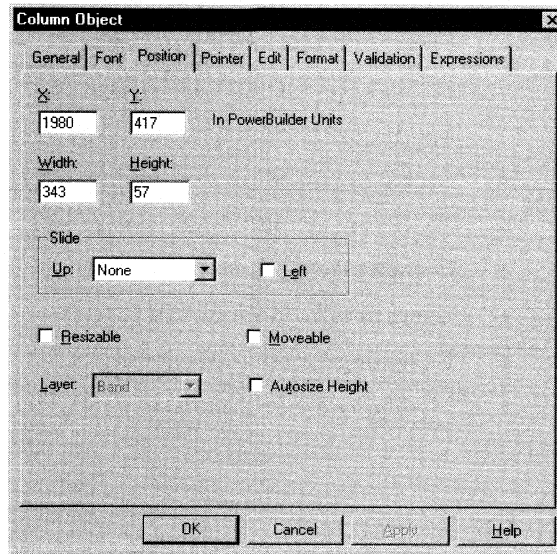
## Sliding objects in a form

You can specify that you want to eliminate blank lines or spaces in a form by sliding columns and other objects to the left or up if there is blank space. You can use this feature to remove extra spaces between fields (such as first name and last name) when you run the form.



❖ **To use sliding columns or objects:**

- 1 Select Properties from the object's popup menu and then select the Position page:



- 2 Select the Slide options you want.

Option	Description
Left	Slide the column or object to the left if there is nothing to the left. Be sure the object does not overlap the object to the left. Sliding left will not work if the objects overlap
Up - All Above	Slide the column or object up if there is nothing in the row above (the row above must be completely empty for the column or object to slide up)
Up - Directly Above	Slide the column or object up if there is nothing <i>directly above it</i> in the row above

**If you are sliding columns up**

Even blank columns have height, so if you want columns to slide up, you need to specify as Autosize Height all columns above that might be blank and that you want to slide other columns up through. You also specify Autosize Height on the Position property page.

## Modifying general form properties

This section describes the general form properties that you can modify in the workspace:

- ◆ "Specifying a title for a form" next
- ◆ "Setting colors for a form" on page 583
- ◆ "Specifying the display of scrollbars for a form" on page 584
- ◆ "Specifying pointers for a form" on page 585
- ◆ "Modifying text in a form" on page 586
- ◆ "Defining the tab order in a form" on page 587
- ◆ "Using borders in a form" on page 589
- ◆ "Prompting for retrieval criteria in a form" on page 589
- ◆ "Modifying the data source of a form" on page 592

### Specifying a title for a form

You can specify a title that displays in the form's title bar when you run a form:

The screenshot shows a window titled "Employee Data" with a title bar containing a question mark icon, the text "Employee Data", and standard window control buttons. The form content is as follows:

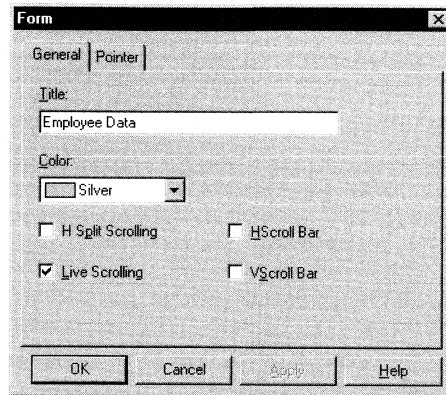
Employee ID: 102	Birth Date: 06/05/1958
Manager ID: 501	Soc. Sec. No.: 017-34-9033
Emp. First Name: Fran	Salary: \$45,700.00
Emp. Last Name: Whitney	Start Date: 02/26/1986
Department ID: 100	Termination Date: 00/00/0000
Street: 49 East Washington Street	Status: <input checked="" type="radio"/> Active
City: Needham	<input type="radio"/> Terminated
State: MA	<input type="radio"/> On Leave
Zip Code: 02192-	Health Insurance: <input checked="" type="checkbox"/>
Phone: (617) 555-3985	Life Insurance: <input checked="" type="checkbox"/>
Sex: <input type="radio"/> Male	Day Care: <input type="checkbox"/>
<input checked="" type="radio"/> Female	

At the bottom of the window, the status bar shows "Ready [ Pre-Release ]" and "Nothing Selected".

❖ **To specify a title for the form:**

- 1 Display the form's popup menu and select Properties.

The Form property sheet displays:



- 2 On the General page, specify the title in the Title box and click OK.

**To change a form's name in the current library (PBL)**

To change a form's name in the current library, select File>Save As from the menu bar, name the form, add a comment, and click OK.

## Setting colors for a form

You can set different colors for each element of a form to enhance the display of information.

❖ **To set colors for a form:**

- ◆ Do one of the following:

To set colors for	Do this
The form's background	Position the pointer on an empty spot in the form, display the popup menu, then select Properties. On the General page of the Form property sheet, select a color from the Color dropdown listbox

To set colors for	Do this
An object	Select the object and use the Foreground Color dropdown toolbar and the Background Color dropdown toolbar  Alternatively, you can position the mouse pointer on the object, display the popup menu, then select Properties. For objects that use text, you can set colors for background and text on the Font page of the property sheet. For drawing objects, you can set colors on the General page of the property sheet

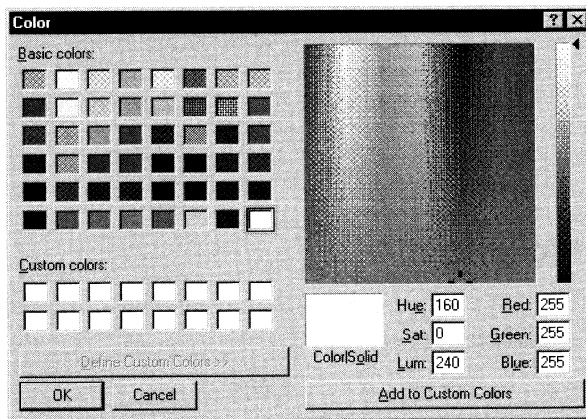
## Defining your own colors

You can define your own custom colors for use in forms and reports.

❖ **To maintain your custom colors:**

- 1 Select Design>Custom Colors from the menu bar.

The Color dialog box displays:



- 2 Define your custom colors.

**FOR INFO** For information about working in the Color dialog box to define custom colors, see "Defining colors" on page 38.

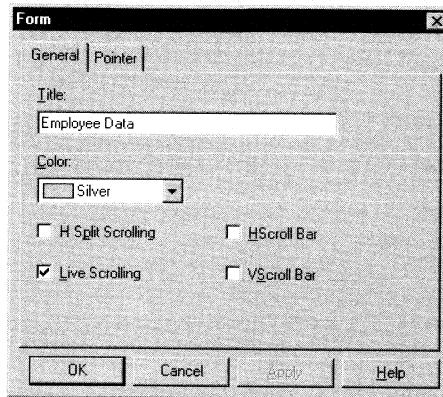
## Specifying the display of scrollbars for a form

You can specify whether your form has scrollbars when you run it.

❖ **To specify scrollbars for a form:**

- 1 Display the form's popup menu and select Properties.

The Form property sheet displays:



- 2 On the General page, select the type of scrollbar you want and click OK.

## Specifying pointers for a form

You can specify a particular pointer image to display when the mouse pointer is over a specific area of a form.

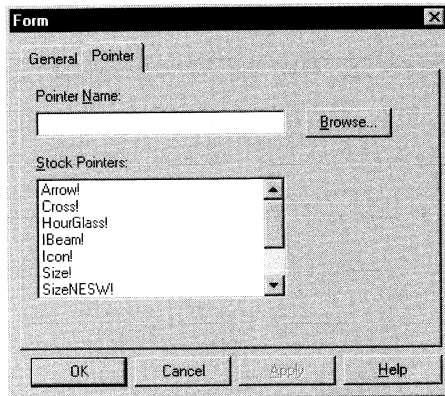
❖ **To change the mouse pointer used while running the form:**

- 1 Position the pointer over the element of the form whose pointer you want to define, display the popup menu, and select Properties to display the property sheet.

You can set a pointer for the entire form and for specific objects.

- 2 Select the Pointer page.

Here's the Pointer property page for the entire form:



- 3 Choose the pointer from the Stock Pointers list, or, if you have files containing pointer definitions (CUR files), enter a pointer filename. You can use the Browse button to search for the file.
- 4 Click OK to return to the workspace.

## Modifying text in a form

When InfoMaker initially generates the basic form, it uses the following:

- ◆ For the text and alignment of column headings and labels, InfoMaker uses the column's extended attributes—definitions created in the Database painter by selecting Properties from a column's popup menu
- ◆ For fonts, InfoMaker uses the properties of the table—definitions created in the Database painter by selecting Properties from the table's popup menu

You can override any of these defaults in a particular form.

### ❖ To modify text in a form:

- 1 Select the text.

The first box in the StyleBar is now active:

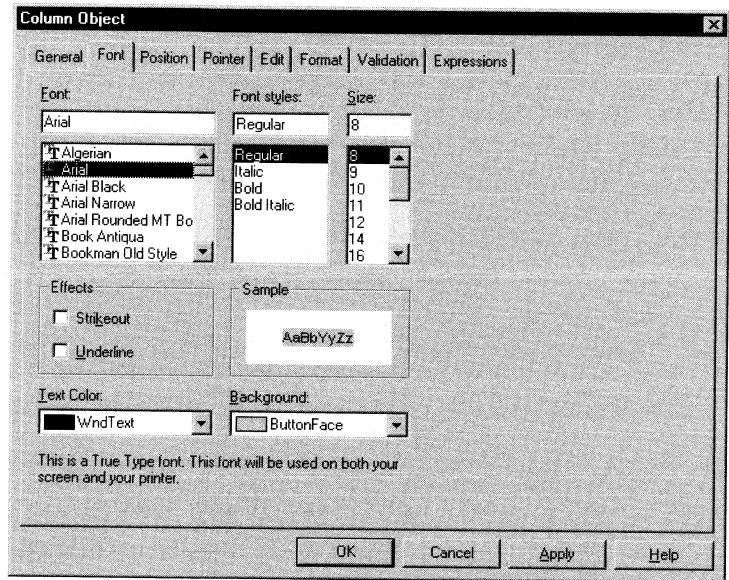


- 2 Type the new text.

Use ~n~r to start a new line in the text. For example, typing Employee~n~rFirst Name will place First Name on the next line.

❖ **To change the text properties for an object in a form:**

- 1 Select the object.
- 2 Do one of the following:
  - ◆ Change the text properties in the StyleBar.
  - ◆ Select Properties from the object's popup menu, select the Font page, and then change the font properties as needed.



## Defining the tab order in a form

When InfoMaker generates the basic form, it assigns controls (including data columns) to a default **tab order**, the default sequence in which focus moves from control to control when you press the TAB key when running the form. InfoMaker assigns tab values in increments of 10 in left-to-right and top-to-bottom order.

---

**Tab order is not used in the workspace**

Tab order is used only when a form is run, not in the workspace.

---

---

**On Macintosh**

On the Macintosh, the TAB key changes the focus among the edit controls and listboxes on a form. Other controls, such as command buttons, never have focus.

---

You can change the tab order.

❖ **To change the tab order in a form:**

- 1 Click the Tab Order button in the PainterBar.

*or*

Select Design>Tab Order from the menu bar.

The current tab order displays.

- 2 Use the mouse or the TAB key to move the pointer to the tab value you want to change.
- 3 Enter a new tab value (0-9999).

The value 0 removes the control from the tab order so you cannot tab to the control. It doesn't matter what value you use (other than 0); all that matters is relative value. For example, if you want to tab to column B after column A but before column C, set the tab value for column B so it is between the value for column A and the value for column C.

- 4 Repeat the procedure until you have the tab order you want.
- 5 Click the Tab Order button in the PainterBar again.

*or*

Select Design>Tab Order from the menu bar again.

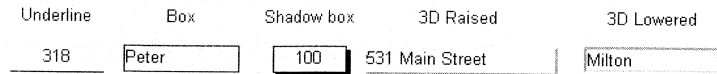
InfoMaker saves the tab order and the tab-order display turns off.

Each time you select Tab Order, InfoMaker reassigns tab values to include any controls that have been added to the form and to allow space for inserting new controls in the tab order.



## Using borders in a form

You can place borders around objects to enhance their appearance. InfoMaker provides five types of borders in forms:



### ❖ To add a border to an object in a form:

- 1 Select the objects you want to add a border to.
- 2 Click the button for the border you want in the Borders dropdown toolbar in the PainterBar.

InfoMaker places the border around the selected objects.

For example, to add a raised border to the objects, click the Raised button in the Borders dropdown toolbar.

---

### Using the object's property sheet for borders

You can also add or modify an object's border by selecting Properties from the object's popup menu and selecting a border from the Border dropdown listbox on the General property page.

---

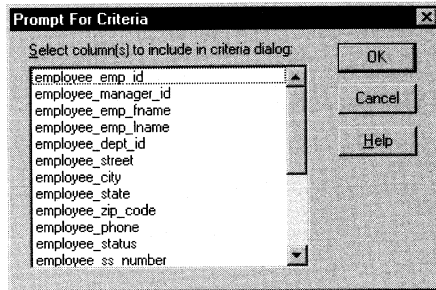
## Prompting for retrieval criteria in a form

You can define your form so that it always prompts you for retrieval criteria just before it retrieves data. In this way, you limit the data retrieved.

❖ **To prompt for retrieval criteria in a form:**

- 1 Select Design>Prompt for Criteria from the menu bar.

The Prompt for Criteria dialog box displays listing all columns in the form:



- 2 Select the columns for which you want to specify retrieval criteria when you run the form.

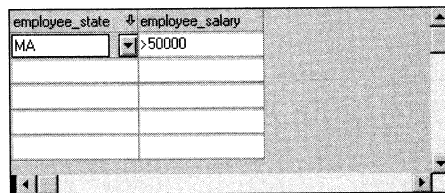
Selected columns are highlighted.

- 3 Click OK.

What happens

If you have specified prompting for criteria, InfoMaker displays the Specify Retrieval Criteria dialog box when you run the form—just before the retrieval is done.

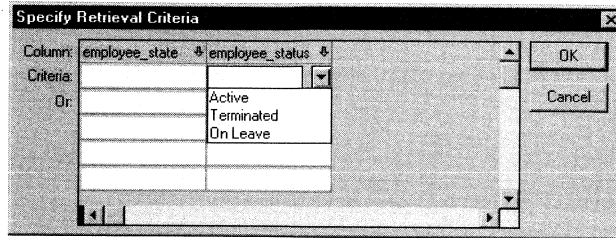
Each column you selected in the Prompt for Criteria dialog box displays in the grid. For example, if you selected the State column and the Salary column, you can specify criteria so only data for employees in Massachusetts with salaries greater than \$50,000 is retrieved:



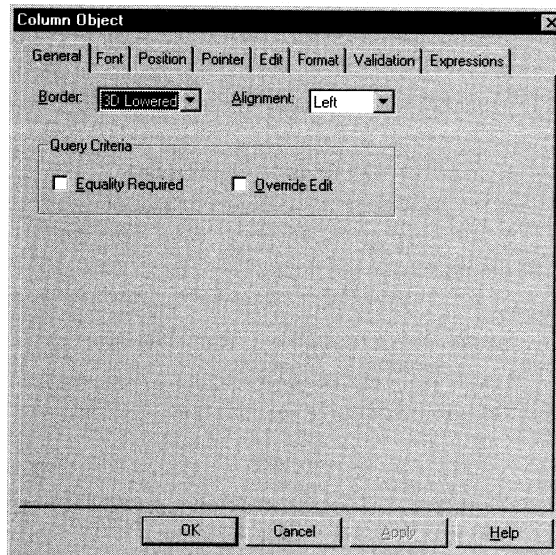
You specify criteria exactly as you do in the grid with the Quick Select data source. Criteria specified are added to the WHERE clause for the SQL SELECT statement defined for the form.

## Using edit styles

If a column uses a code table or the RadioButton, CheckBox, or DropDownList edit style, an arrow displays in the column header, and you can select a value from a dropdown listbox when you run the form.



If you don't want the dropdown listbox used for a column when you specify retrieval criteria, select Properties from the column's popup menu and select the Override Edit checkbox on the General page in the Column Object property sheet:



## Forcing the entry of criteria

If you have specified prompting for criteria for a column, you have the option of entering criteria when you run the form. If you want, you can make the entry of criteria a requirement rather than an option. For example, you might have a departmental form that requires the selection of a department.

You can force the entry of criteria for a column by selecting the Equality Required checkbox on the General page in the Column Object property sheet.

When you run the form, InfoMaker underlines the column header in the grid. You must enter criteria for the column. If the column values are presented in a dropdown listbox, you must pick one. If the column values are numeric, you must use the = operator and not other operators, such as < or >=.

**FOR INFO** For information about how you can specify selection criteria in the grid, see "Specifying selection criteria" on page 161.

## Modifying the data source of a form

When modifying a form, you might realize that you haven't included all the columns you need. Or you might need to define retrieval arguments. You can modify the data source from the Form painter workspace by graphically modifying the SQL SELECT statement.

### ❖ To modify the form's data source:

- 1 Select Design>Edit Data Source from the menu bar.

*or*

Point at the workspace and select Edit Data Source from the popup menu.

---

#### **If your form style is Master/Detail**

For a master/detail many-to-one or one-to-many form, the Select Data Source dialog box displays. Select the data source you want to modify: the master or the detail.

---

InfoMaker returns you to the Select painter.

If you used Quick Select to define the data source, this might be the first time you have seen the Select painter.

**FOR INFO** For information, see "Using SQL Select" on page 166.

- 2 Modify the SELECT statement graphically using the same techniques you used when creating it.
- 3 Click the SQL Select button to return to the workspace.  
*or*  
Select Design>Data Source from the menu bar.

**Changing the table**

If you change the table referenced in the SELECT statement, InfoMaker maintains the columns in the workspace (now from a different table) only if they match the data types and order of the columns in the original table.

---

**Adding columns**

You can add columns to the SELECT statement. However, the column does not automatically display in the workspace. You need to add columns manually, one at a time.

**❖ To use a new column in a form:**

- 1 Click the Column button in the Controls dropdown toolbar.  
*or*  
Select Controls>Column from the menu bar.
- 2 Click where you want to place the column.  
The Select Column dialog box displays.
- 3 Select the new column and click OK.

## Adding objects to the form

The following topics tell you how to enhance your form by adding objects:

- ◆ "Adding columns to a form" next
- ◆ "Adding text to a form" on page 594
- ◆ "Adding computed fields to a form" on page 595
- ◆ "Adding pictures to a form" on page 599
- ◆ "Adding command buttons to a form" on page 600
- ◆ "Adding picture buttons to a form" on page 603
- ◆ "Adding reports to a form" on page 605
- ◆ "Adding drawing objects to a form" on page 606

## Adding columns to a form

You can add columns to a form: you can restore columns you have deleted or add columns after you have modified the data source to include more columns.

❖ **To add a column to a form:**

- 1 Click the Column button in the Controls dropdown toolbar.  
*or*  
Select Controls>Column from the menu bar.
- 2 Click where you want to place the column.  
The Select Column dialog box displays, listing columns not currently in the form.
- 3 Select the column and click OK.

The column is added. No header or label is added, but you can create a header or label by adding a text object to the form.

## Adding text to a form

When InfoMaker generates a basic form from a form style and data source, it places columns and their headings in the workspace. You can add text anywhere you want to make the form easier to understand.

❖ **To add text to a form:**

- 1 Click the Text button in the PainterBar.

*or*

Select Controls>Text from the menu bar.

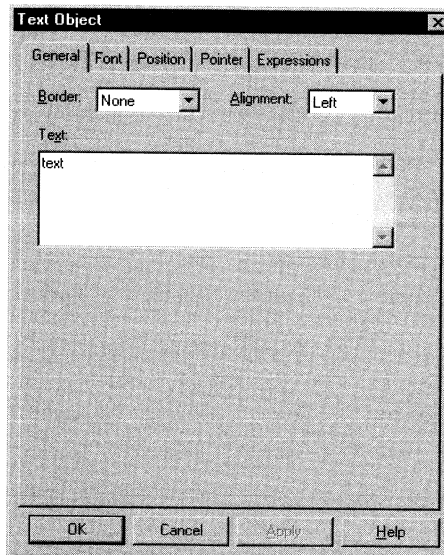
- 2 Click where you want the text.

InfoMaker places the text object in the workspace and displays the word *text* on the object and in the Text box in the StyleBar.

- 3 In the text box, select the word *text* and type the new text.

*or*

Select Properties from the text object's popup menu to display the Text Object property sheet, select the word *text*, and type the new text in the Text box on the General page:



- 4 (Optional) Change the font, size, style, and alignment for the text by using the StyleBar or changing the properties on the Font page in the property sheet.

## Adding computed fields to a form

You can use computed fields to perform calculations in the form. Typical uses of computed fields include:

- ◆ **Calculations based on column data that change for each retrieved row** For example, if you are retrieving yearly salary, you could define a computed field that displays monthly salary (defined as `Salary / 12`).
- ◆ **Summary statistics** For example, you can use a computed field to calculate the average salary of all the retrieved rows.
- ◆ **Concatenated fields** For example, if you are retrieving first name and last name, you can define a computed field that concatenates the values so they appear with only one space between them (defined as `Fname + " " + Lname`).
- ◆ **System information** For example, you can place the current date and time in a form by using computed fields (defined as `Today( )` and `Now( )`).

## About defining computed columns and computed fields

When creating a form, you can define computed columns and computed fields:

- ◆ In the Select painter, you can define computed columns when you are defining the SELECT statement that will be used to retrieve data into the form
- ◆ In the Form painter workspace, you can define computed fields *after* you have defined the SELECT statement

The difference  
between the two ways

When you define a computed column in the Select painter, the column's definition is part of the SELECT statement and the value is calculated *by the DBMS* when the data is retrieved. The computed column's value does not change until data has been updated and retrieved again.

When you define a computed field in the Form painter workspace, the value of the column is calculated by *InfoMaker* in the form *after* the data has been retrieved. The value changes dynamically as the data in the form changes.

Recommendation

If you want your DBMS to do the calculations on the server before bringing data down and you don't care about dynamically updating the computed values, define computed columns as part of the SELECT statement.

If you want computed values to change dynamically, define computed fields in the Form painter workspace, as described next.

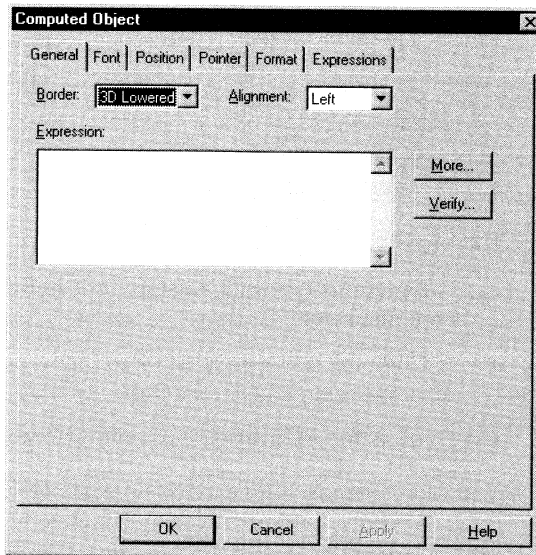


## Defining a computed field

### ❖ To add a computed field to a form:

- 1 Click the Compute button in the Controls dropdown toolbar.  
*or*  
Select Controls>Computed Field from the menu bar.
- 2 Click where you want the computed field.

The Computed Object property sheet displays:



- 3 In the Expression box, enter an expression to define the computed field, click the Verify button to check the expression's validity, modify it as needed, and then click OK.

*or*

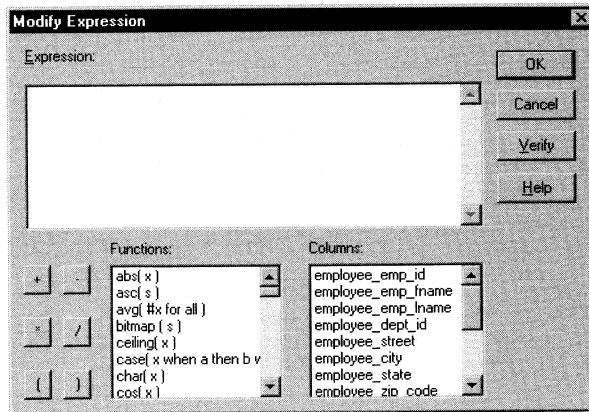
Click the More button to display the Modify Expression dialog box. In this dialog box, you can easily enter a complicated expression. When you are done you can verify it, click OK, and then click OK again to close the property sheet.

**FOR INFO** For more, see "About the Modify Expression dialog box" next.

InfoMaker returns you to the workspace with the computed field in place.

About the Modify Expression dialog box

The Modify Expression dialog box provides you with lists and buttons to help you create the computed field:



The Modify Expression dialog box contains:

- ◆ A Functions box with a list of built-in functions you can use in the computed field
- ◆ A Columns box with a list of columns, named computed fields, and retrieval arguments in the form
- ◆ Buttons for adding operators and parentheses

Entering the expression

You can enter any valid expression when defining a computed field. You can paste operators, columns, existing computed fields, retrieval arguments, and functions into the expression from information in the Modify Expression dialog box. You can use the + operator to concatenate strings.

**Remember**

The expression you are entering is an *InfoMaker* expression; it is not a SQL expression processed by the DBMS. So the expression follows InfoMaker rules.

FOR INFO For complete information about expressions, see Chapter 22, "Operators and Expressions".

Examples

Here are some examples of computed fields:

To display	Enter this expression
Current date	Today( )
Current time	Now( )

To display	Enter this expression
Concatenation of Fname and Lname columns for each row	Fname + " " + Lname
Monthly salary if Salary column contains annual salary	Salary / 12
Four asterisks if the value of the Salary column is greater than \$50,000	IF(Salary > 50000, "****", "")
Average salary of all retrieved rows	Avg(Salary)
Count of retrieved rows, assuming each row contains a value for EmpID	Count(EmpID)

**FOR INFO** For more information about the functions you can use in computed fields in the Form painter, see Chapter 23, "DataWindow Painter and InfoMaker Functions" and online Help.

## Adding pictures to a form

You can place pictures, such as your company logo, in a form to enhance its appearance.

### Tips for using pictures

To display a different picture for each row of data, retrieve a column containing picture filenames from the database. In the Database painter, a column containing picture filenames has the Picture checkbox selected on the Display page in the column's property sheet.

**FOR INFO** For more information, see the section on character columns in "Modifying column properties in the Database painter" on page 84.

To compute a picture name when a form is run, use the Bitmap function in the expression defining a computed field.

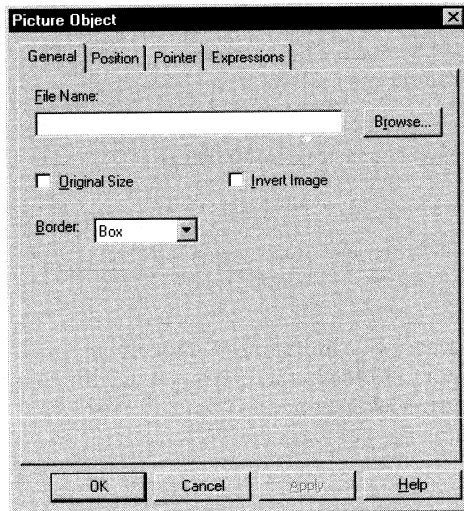
**FOR INFO** For information about the Bitmap function, see Chapter 23, "DataWindow Painter and InfoMaker Functions" and online Help.

### ❖ To add a picture to a form:

- 1 Click the Picture button in the Controls dropdown toolbar.  
*or*  
Select Controls>Picture from the menu bar.

- 2 Click where you want the picture to display.

The Picture Object property sheet displays:



- 3 Click Browse to display the Select Picture dialog box and navigate to find the picture file.

*or*

Enter a filename in the File Name box.

The picture must be a bitmap (BMP) file, a Windows metafile (WMF), or a runlength-encoded (RLE) file.

---

#### **On Macintosh**

On the Macintosh, the picture must be a bitmap (BMP) file or a runlength-encoded (RLE) file.

---

- 4 Click OK.

You return to the workspace with the picture in place.

## **Adding command buttons to a form**

You can add command buttons to a form. Command buttons are used to carry out an action. For example, you can add a button that prints the current form.

❖ **To add a button to a form:**

- 1 Click the Button button in the Controls dropdown toolbar.

*or*

Select Controls>CommandButton from the menu bar.

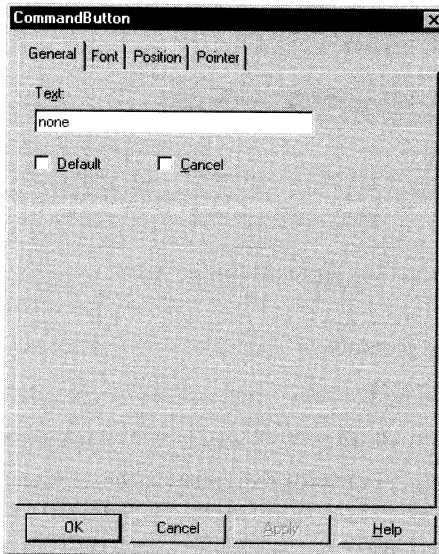
- 2 Click where you want to place the command button.

A button with the text *none* displays. The text *none* also displays in the text box in the StyleBar.

- 3 In the text box in the StyleBar, select the text and type the new text.

*or*

Select Properties from the command button's popup menu to display the CommandButton property sheet, select the text, and type the new text in the Text box on the General page.



- 4 (Optional) Specify the command button as a default or cancel button.

**FOR INFO** For more information, see "Specifying default and cancel buttons" on page 602.

- 5 Click OK.

## Making the command button work

To make the button do something when it is clicked when you run a form, you associate an action with it. Each form style has its own set of actions you can choose from for a button.

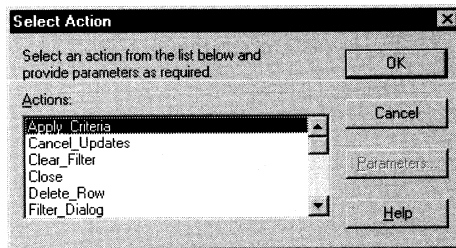
### ❖ To associate an action with a button:

- 1 Move the pointer to the button you added to the form, display the popup menu, and select Action.

*or*

Select the button and select Edit>Action from the menu bar.

The Select Action dialog box displays the actions that are provided with the form style you are using



- 2 Select the action from the list.
- 3 If the action takes parameters, click the Parameters button and supply the parameters.
- 4 Click OK.

You return to the Form painter workspace.

**FOR INFO** For information about the actions provided with the built-in form styles, see "Actions in forms" on page 550.

If you need information about actions provided by a style defined in your organization, talk with the style's developer.

## Specifying default and cancel buttons

You can define a button as the default button in a form and you can define a Cancel button.

**Default button behavior**

If you define a default button, pressing the ENTER key when the focus is not on another button is the same as clicking the default button. If the focus is on another button, pressing ENTER is the same as clicking the button that has focus.

InfoMaker places a bold border around the default button (or the button with focus if you explicitly tab to a button).

**Cancel button behavior**

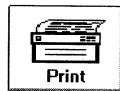
If you define a cancel CommandButton, pressing the ESC key is the same as clicking the cancel button.

❖ **To define a default or cancel button:**

- ◆ Select Properties from the command button's popup menu and then select the Default or Cancel checkbox on the General page.

## Adding picture buttons to a form

Picture buttons are identical to command buttons in their functionality. The only difference is that you can specify a bitmap (BMP) file, a Windows metafile (WMF), or a runlength-encoded (RLE) file to display in the button. Use these controls when you want to be able to represent the purpose of a button using a picture instead of just text.




---

### On Macintosh

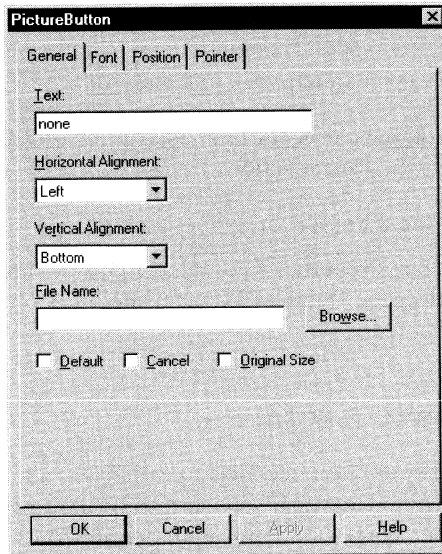
On the Macintosh, you can specify a bitmap (BMP) file or a runlength-encoded (RLE) file to display in the button.

---

❖ **To add a picture button to a form:**

- 1 Click the PictureBox button in the Controls dropdown toolbar.  
*or*  
Select Controls>PictureBox from the menu bar.
- 2 Click where you want the picture button to display.  
A picture button displays in the workspace.

- 3 Select Properties from the picture button's popup menu to display the PictureButton property sheet, select the text, and type the new text in the Text box on the General page:



- 4 Click Browse to display the Select Picture dialog box and navigate to find the picture file.

*or*

Enter a filename in the File Name box.

The picture must be a bitmap (BMP) file, a Windows metafile (WMF), or a runlength-encoded (RLE) file.

---

#### **On Macintosh**

On the Macintosh, you can specify a bitmap (BMP) file or a runlength-encoded (RLE) file to display in the button.

---

- 5 Specify other properties of the button as needed.
- 6 Click OK.

You return to the workspace with the picture in place.

**FOR INFO** For information about associating the picture button with an action, see "Making the command button work" on page 602. The procedure is the same.

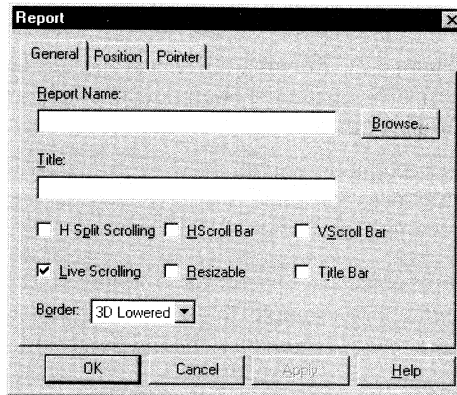


## Adding reports to a form

You can add reports that you created in the Report painter to a form. The report must be in the current library.

### ❖ To add a report to a form:

- 1 Click the Report button in the Controls dropdown toolbar.  
*or*  
Select Controls>Report from the menu bar.
- 2 Click where you want the report to display.  
A report object displays in the workspace.
- 3 Select Properties from the report object's popup menu to display the Report property sheet:



- 4 Click Browse to display the Select Report dialog box and select a report.

*or*

Enter a report name in the Report Name box.

InfoMaker places the report in the form (you see everything but the data, which is displayed when you run the form).

At this point, you may want to resize the report or change some of its other properties. For example, you may want to make the report itself resizable. To change properties, display the report's property sheet again (by double-clicking on the report or using the report's popup menu) and change properties on the appropriate page. The resizable checkbox is on the General page.

## Adding drawing objects to a form

You can add the following drawing objects to a form to enhance its appearance:

- Line
- Oval
- Rectangle
- RoundRectangle

❖ **To add a drawing object to a form:**

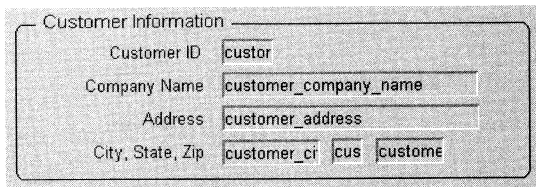
- 1 Click the drawing object in the Controls dropdown toolbar.  
*or*  
Select Controls>Line, Controls>Oval, Controls>Line, Controls>Rectangle, Controls>RoundRectangle, from the menu bar.
- 2 Click where you want the object to display.
- 3 Resize or move the drawing object as needed.
- 4 Change the drawing object's properties as needed by selecting Properties from the popup menu and working on the appropriate property page.

For example, you might want to specify a fill color for a rectangle or thickness for a line. To do so, you work on the General page in the rectangle or line object's property sheet.

Using drawing objects for grouping

Drawing objects are useful for grouping objects in a form or providing design highlights. For example, to group some objects you can place a colored rectangle behind them by selecting Send To Back from the rectangle's popup menu.

Here are some objects grouped with a round rectangle and a text label:



Customer Information

Customer ID	custor
Company Name	customer_company_name
Address	customer_address
City, State, Zip	customer_ci   cus   custome

❖ **To group objects with a round rectangle and text in the Form painter:**

- 1 Select Controls>RoundRectangle from the menu bar and add a round rectangle to the workspace.
- 2 Size the rectangle and place it over the objects you want to group.

- 3 Select **Send To Back** from the rectangle's popup menu to place the rectangle behind the group of objects.
- 4 Click the **Text** button in the **Controls** dropdown listbox, add a text object to the workspace, and place it on the round rectangle.
- 5 Select a background color for the text object from the **Background Color** dropdown toolbar and select the color, in this case gray to match the rectangle.
- 6 Position all objects as needed.

## Highlighting information in a form

Every object in a form has a set of properties that determine basically what the object looks like and where it is located. For example, the values in a column of data display in a particular font and color, in a particular location, with or without a border, and so on.

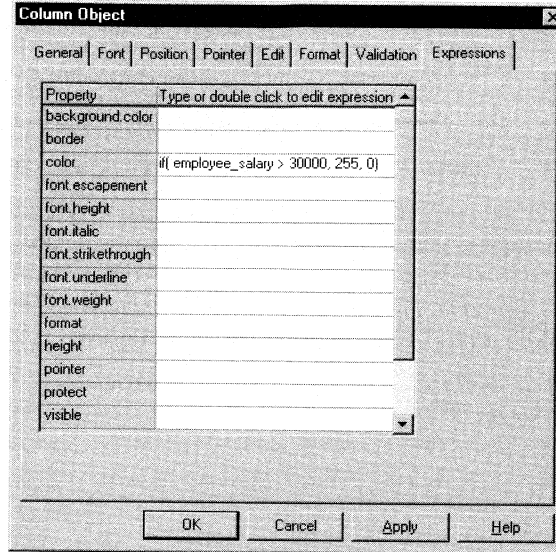
This chapter thus far has described how you modify various properties of objects in the workspace. These modifications are static. They don't change when you run the form.

In contrast, you can also tell InfoMaker to modify some of these properties when you run the form *based on conditions you specify*. These modifications are dynamic. They are based on information only available when you run the form.

For example, you can tell InfoMaker to show salaries greater than \$30,000 in red. When you run the form, salaries greater than \$30,000 display in red. When you enter new salary data in the form, a salary greater than \$30,000 displays in red after you enter the salary and then move the pointer to or tab to another column of data.

You modify properties based on conditions you specify by selecting Properties from an object's popup menu and entering an expression (which usually uses the If function) on the Expressions property page. In this example, the expression for the salary column's color property is:

```
if( employee_salary > 30000, 255, 0 )
```



For more information

For information on modifying properties based on conditions you specify, see Chapter 8, "Highlighting Information in Reports and Forms". Typing an expression on the Expressions property page is done in the same way in forms and reports.

## **Displaying and validating data in a form**

Some of the most important ways you can enhance forms are to create:

- ◆ Display formats to specify how values are formatted
- ◆ Edit styles to specify styles in which values are presented and entered
- ◆ Validation rules to ensure that values entered in forms are valid

Display formats, edit styles, and validation rules are usually created in the Database painter (if you have access to the database). In the Form painter, the database definitions apply to the data unless you override them with new definitions. The new definitions you create in the Form painter do not affect the definitions in the database; the definitions are used only in the form.

**FOR INFO** For more information, see Chapter 6, "Displaying and Validating Data".

PART 5

# Applications

This part describes how to create and deploy InfoMaker applications.





**About this chapter**

You can bundle reports, forms, and data pipelines into a package to create a reporting and database-maintenance application. This chapter describes how to create an InfoMaker application.

**Contents**

<b>Topic</b>	<b>Page</b>
About applications	614
Creating an application	616
Reusing an application	622
Running an application	624
Using a pipeline in an application	630

## About applications

When you have completed reports, forms, or data pipelines, you can bundle them in an executable file to create a reporting and database-maintenance application. The reports, forms, and pipelines in an application are usually related by topic, but they need not be. You can bundle any reports, forms, and pipelines you want.

Where you create an application

In InfoMaker, you create applications in the Environment painter. In the Environment painter you select the reports, forms, and pipelines you want to include in your application and then you create the application.

---

### No queries

InfoMaker applications do not include queries, which also display in the Environment painter.

---

Identifying an application

Most applications can be identified by selecting Help>About to display information about the application such as the name, name of the company producing the application, version, and so on. After you create an application, you can modify the application's initialization file so your users will see customized information.

**FOR INFO** For information about identifying your application, see "Identifying your application" on page 625.

Running an application

You can run the application the same way you run other Windows 95 applications. After you run the application, you can run the reports and forms and pipelines included in the application.

Using pipelines in an application

Having the ability to execute pipelines in an application is particularly useful for mobile users working on laptops that are often not connected to a corporate database. You can pipe a corporate database to the laptop, use forms to update the database, and run reports against the local database. Then you can pipe the local data to the corporate database.

**FOR INFO** For information about executing a pipeline in an application and modifying the pipeline definition from the application, see "Using a pipeline in an application" on page 630.

Distributing an application

An application can be for your personal use only, but you can also distribute it to others to use. Users can run your application outside the InfoMaker environment. Having InfoMaker installed is not required.

The requirements for distributing an application with a pipelines differ from the requirements for distributing an application with only forms and reports because pipelines can modify the database by adding and dropping tables.

**FOR INFO** For information about distributing InfoMaker applications to others, see Chapter 18, "Deploying Your Application on Windows", and Chapter 19, "Deploying Your Application on the Macintosh"

## Creating an application

When you create an application, all the reports, forms, and pipelines you include must be in one library (PBL). In the Environment painter, you can move or copy reports, forms, and pipelines from one library to another as needed.

**FOR INFO** For information about moving or copying objects, see Chapter 2, "Managing Your Environment".

---

### **Remember to refresh the library list**

If you've just created new reports, forms, or pipelines, remember to refresh the library list by selecting Design>Refresh from the menu bar. Doing this makes all objects in the current library display.

---

#### ❖ **To create an application:**

- 1 Connect to the database that the executable will be using.
- 2 Open the Environment painter.  
  
The objects in the current library (PBL) display. If you don't see the objects you want, open another library.
- 3 Select the reports, forms, and pipelines you want to package in the application. Also select any DropDownDataWindows reports (used for DropDownDataWindow edit styles) and reports nested in the reports or forms you have selected.

---

### **Object comments may be included in the application**

The comments you provide for objects in the Environment painter may be included in the application by default under certain circumstances (see step 8 and step 9). Comments that are understandable for a user of the application make it easier for the user to select from lists of objects.

---

Press the CTRL key while selecting entries individually. To select a group of entries, select one entry, press SHIFT, and click another entry to select all entries between the two.

---

### **On Macintosh**

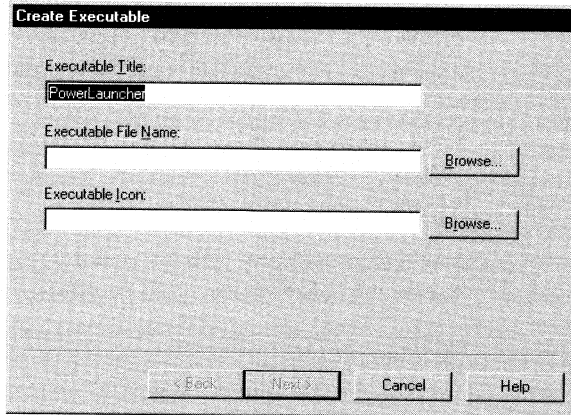
On Macintosh, press the COMMAND key while selecting entries individually. To select a group of entries, select one entry, press SHIFT, and click another entry to select all entries between the two.

---

- 4 Click the Create Executable button.  
*or*

Select Entry>Create Executable from the menu bar.

If you are creating your first application in the current library, the Create Executable dialog box displays.



If you have created applications in the current library before, you can reuse your most recent application and avoid starting from scratch.

FOR INFO For information, see "Reusing an application" on page 622.

- 5 Name your application's executable file by filling in the Executable File Name box.

This is the name of the executable file that will be created. You can use a short or long filename with or without spaces. On Windows, be sure to retain the EXE file extension.

---

#### On Macintosh

The application name can be up to 31 characters, including spaces.

---

The default location is where InfoMaker is installed. You can click the Browse button if you want to specify a different location for the executable file in the Select Executable File dialog box.

---

**To quickly prototype an application**

Steps 6–11 describe how to select an icon for your application and define text, MicroHelp, and pictures for the application's toolbar buttons. You can create an application quickly by stopping here, clicking OK now, not selecting an icon when prompted, and clicking OK again in the Executable Items dialog box. By doing this, you can create a prototype application to see if it's what you want before you enhance it. Then you can reuse your prototype and enhance it.

---

- 6 Click the Browse button next to the Executable Icon box to assign an icon to the executable file.

The Select Icon dialog box displays.

The icon you choose will be used if you create a shortcut for the executable file and when you minimize the executable.

---

**On Macintosh**

You can use the icon from another file on your Macintosh: copy the icon from that file's Info window and paste it onto the Info window of your application.

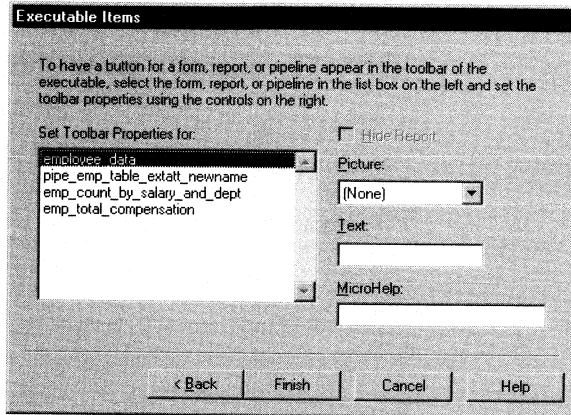
---

- 7 Navigate to an ICO file and click Open.

You return to the Create Executable dialog box.

## 8 Click Next.

The Executable Items dialog box displays, listing all reports, forms, and pipelines you have selected for the executable file:



In the Executable Items dialog box, you can specify a toolbar button for any of the reports, forms, and pipelines in the executable. If you add a report, form, or pipeline button to the toolbar, you can run the report, form, or pipeline by simply clicking the toolbar button.

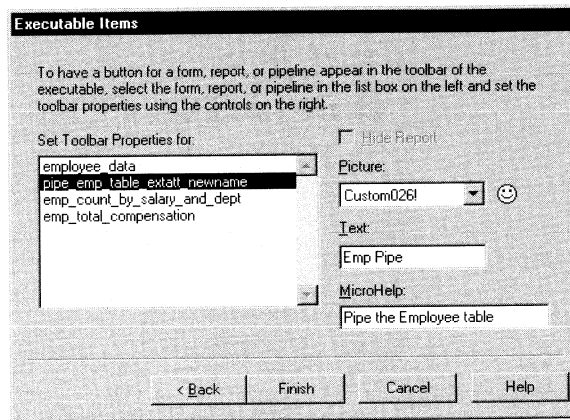
By default, InfoMaker creates an Objects menu in the application that lists all reports, forms, and pipelines. If you do not specify button text for an object, the object name (that you see in the Environment painter) displays in the Objects menu.

InfoMaker will also create a File menu in the application and you can select File>Open Form, File>Open Report, or File>Open Pipeline and then select a form, report, or pipeline from the list that displays. If you do not specify MicroHelp text for an object, the object comments (if any were created in the Environment painter) display in the lists.

9 If you want a toolbar button for a report, form, or pipeline, specify a picture for the button, the text you want to show on the button, and MicroHelp text that displays when the button is clicked as shown in the following:

- 1 Select the report, form, or pipeline in the Executable Items dialog box.
- 2 In the Picture dropdown listbox, select the picture you want to display in the button for the report, form, or pipeline. The image you choose displays next to the listbox.

- 3 In the Text box, type the text you want to display in the button for the report, form, or pipeline. Up to 14 characters will display on a button.
- 4 In the MicroHelp box, type the MicroHelp text you want to display when the user of the executable selects the button:



Most of the time you should provide text for a button. If no button text is provided, the object name (that you see in the Environment painter) will automatically be used as the button text and in the list of objects in the Objects menu and usually the object name is not appropriate.

If you want an object name or comment (that you see in the Environment painter) to display in the application's Objects menu, in the File>Open *Object* menu, or in MicroHelp, use the following table to decide whether to enter button text or MicroHelp text in the Executable Items dialog box:

Enter button text?	Enter MicroHelp text?	Text In Objects menu, the button, and the name in the File>Open <i>Object</i> dialog box	Text in MicroHelp and the description in the File>Open <i>Object</i> dialog box
No	No	Object name	Object comment
No	Yes	Object name	MicroHelp text
Yes	No	Button text	Object comment
Yes	Yes	Button text	MicroHelp text

- 10 If you want to omit a report from the list of reports in the generated application so users can't run them as standalone reports, select the report and then select the Hide Report checkbox. This is useful for reports that have been included only to support a:



- ◆ DropDownDataWindow edit style
- ◆ Report added to a form
- ◆ Report nested in another report

11 When you have defined all the toolbar items, click Finish.

What happens

InfoMaker creates two files:

- ◆ The executable file with the name you supplied.
- ◆ An initialization file with the same name. For example, if you created the executable file SALES.EXE, InfoMaker also creates a file named SALES.INI. The initialization file records which database is used by the executable file.

FOR INFO For information about running an application, see "Running an application" on page 624.

Identifying your application

By default, InfoMaker includes Powersoft-brand information to identify your application when a user displays the Help>About window. But you can make modifications to the application's initialization file to customize the information that identifies your application.

FOR INFO For information, see "Identifying your application" on page 625.

An application includes a query governor

InfoMaker includes a query governor in an InfoMaker application automatically. The query governor lets you specify the maximum number of rows to be retrieved and the maximum time for retrieval.

FOR INFO For more information, see "Using the query governor in an application" on page 628.

## Reusing an application

When you create an application, you select reports, forms, and pipelines you want to include and click the Create Executable button. When you do this, if an application was previously created in the current library, InfoMaker remembers the most recent one. You can start fresh or reuse the most recent application.

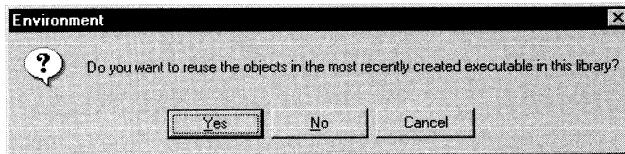
❖ **To reuse your most recent application without adding additional reports, forms, and pipelines:**

- ◆ Press CTRL+click to clear the selected object *before* you click the Create Executable button.  
*or*  
Select an object *that is contained in* your most recent application *before* you click the Create Executable button.

❖ **To add additional reports, forms, and pipelines to your most recent application:**

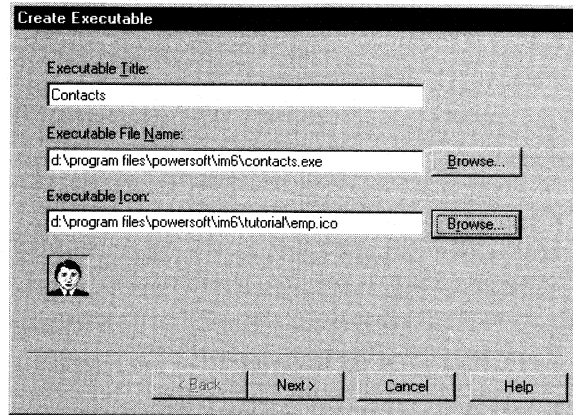
- ◆ Select one or more new objects and click the Create Executable button.

After you click the Create Executable button, InfoMaker displays the following dialog box:



If you want to	Do this	InfoMaker does this
Start from scratch (not add selected objects to your most recent application)	Click No	Includes only your current selections in a new application
Add selected objects to your most recent application	Click Yes	Includes the reports, forms, and pipelines you selected with your most recent application. If these were already in your most recent application, you work with that application again with no additions

When you click No, the Create Executable dialog box displays and you define your new application. When you click Yes, the Create Executable dialog box displays with your most recent application's title and filename and icon. You can rename the application if you want, and continue to define the application:



## Running an application

You can run your application the same way you run other Windows 95 applications. For example, you can double-click the executable file in Explorer or File Manager or you can create an application shortcut on the desktop and double-click the shortcut.

If you create an application shortcut, before you can run your application by double-clicking the shortcut you need to modify the shortcut's Start In property.

---

### **If you are using Windows 3.1**

Instead of creating a shortcut, you can add the application to the Powersoft Program Group (or another program group).

---

#### ❖ **To modify the Start In property of the application shortcut:**

- 1 Position the pointer on the icon, display the popup menu, and select Properties.  
InfoMaker displays the shortcut's property sheet.
- 2 Select the Shortcut tab and type the location of the Powersoft system modules in the Start In box.

---

### **About the location of Powersoft system modules**

When you install InfoMaker, the installation process automatically puts the DLLs in a system folder. If you have changed the names of the folders used for installing, you will need to use your names in this step.

**For Windows 95 and Windows NT 4.0** The folder path is *c:\Program Files\Powersoft\Shared*.

On Windows 95, you must specify the name of the drive. If you use backslashes to specify a relative pathname, the application will be unable to locate the system modules.

**For Windows NT 3.51 and Windows for Workgroups** The folder path is *c:\pwrs\shared*.

**On Macintosh** The folder path is *System Folder:Preferences*.

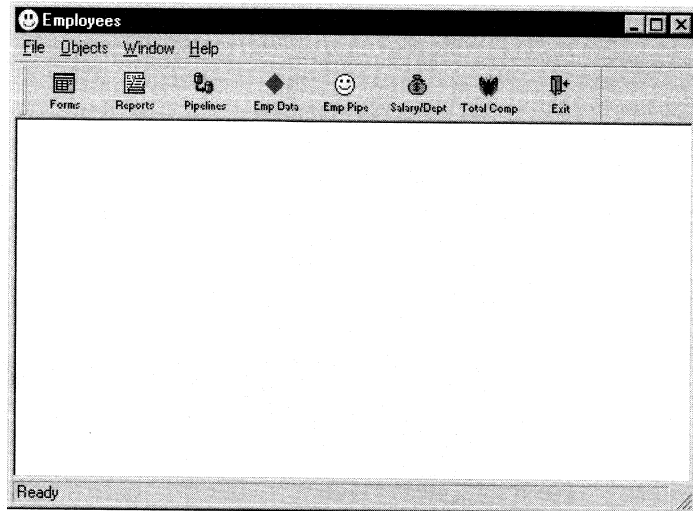
---

### 3 Click OK.

Now you can double-click the application shortcut to run the application.

What happens when you run an application

When you run the application, the application reads information in the application's initialization file and connects to the database to access the data. Then the application's main window displays with a menu bar and toolbar:



The Reports, Forms, Pipelines, and Exit buttons are automatically included in the toolbar. Other buttons are in the toolbar if you defined toolbar buttons for any reports, forms, or pipelines when you created the application.

## Identifying your application

Most applications can be identified by selecting Help>About to display information about the application such as the name, name of the company producing the application, version, and so on. By default, InfoMaker includes the application name and Powersoft-brand information to identify your application.

You can modify the application's initialization file so your users will see customized information about you and your company.

❖ **To customize the application-identity information users see in the Help>About window:**

- 1 Open the application's initialization file and create an About section by adding the following line:

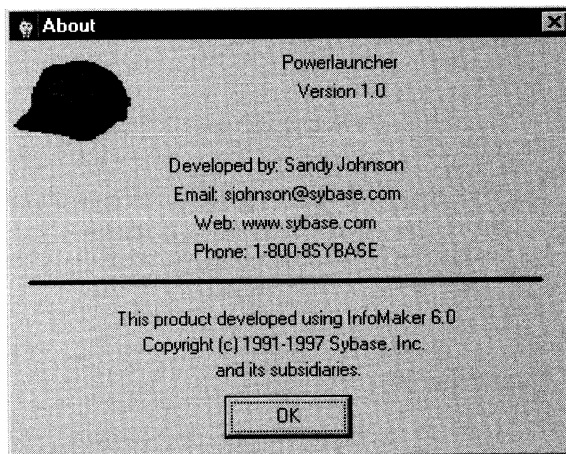
```
[About]
```

- 2 Include any or all of lines like the following in the About section:

```
[About]
developer_name = Sandy Johnson
software_version = 1.0
company_logo = internet.bmp
email_address = sjohnson@sybase.com
web_address = www.sybase.com
phone_number = 1-800-8SYBASE
```

- 3 Save the initialization file.

Now when you open the Help About window, you see the customized information in the Help>About window:



## Changing the application icon on Macintosh

You can select another icon already on your Macintosh for your application.

❖ **To change the application icon:**

- 1 Select the file that has the icon you want to use for your application.

- 2 Select File>Get Info from the menu bar.  
*or*  
Press COMMAND +I.  
An Info window displays with the icon on top.
- 3 Select the icon and copy it.
- 4 Select your application icon in the folder in which you created the application.
- 5 Select File>Get Info  
*or*  
Press COMMAND +I.  
An Info window displays the application icon at the top.
- 6 Select the icon and paste the clipboard contents.  
The new icon displays for your application.

## Running a report, form, or pipeline

To run a report, form, or pipeline, you can do any of the following:

- ◆ Click it in the toolbar if you defined a toolbar item for it.
- ◆ Select it from the Objects menu, which lists all reports, forms, or pipelines in the executable.

If you specified a toolbar button and button text for a report, form, or pipeline, this text displays instead of the report, form, or pipeline object name in the Objects menu.

- ◆ Click the Reports, Forms, or Pipelines button in the toolbar and select the item in the resulting dialog box.

What happens when you run a report or form

Running a report or form in an executable file is the same as running a form or previewing a report in InfoMaker.

---

### When a report is retrieving data

The rows retrieved displays in MicroHelp and the retrieve button changes to a Cancel button. You can click the Cancel button at any time to stop retrieval.

---

What happens when you run a pipeline

When you run a pipeline in an application, the Pipeline window displays. You can then execute the pipeline manually or specify delayed execution. Advanced users of the application can redefine a pipeline object if you instruct users how to do this.

**FOR INFO** For information about executing a pipeline and redefining a pipeline object in the application, see, "Using a pipeline in an application" on page 630.

For more information

For information about previewing reports, see "Previewing a report" on page 205.

For information about running forms that use built-in InfoMaker form styles, see "Working with forms" on page 546.

For information about running forms using a style created by a PowerBuilder developer at your organization, see the developer.

## **Managing the toolbar**

You can move the toolbar and suppress the display of text in the toolbar by selecting items from the popup menu.

## **Managing the open reports, forms, and pipelines**

You can manage the reports, forms, and pipelines you have opened by selecting items from the Window menu, which InfoMaker automatically provides in the executable.

## **Using the query governor in an application**

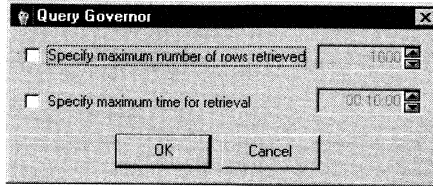
When an InfoMaker application (executable file) is created, a query governor is included automatically.



❖ **To access and use the query governor in an application:**

- 1 In the running InfoMaker application, select File>Query Governor from the menu bar.

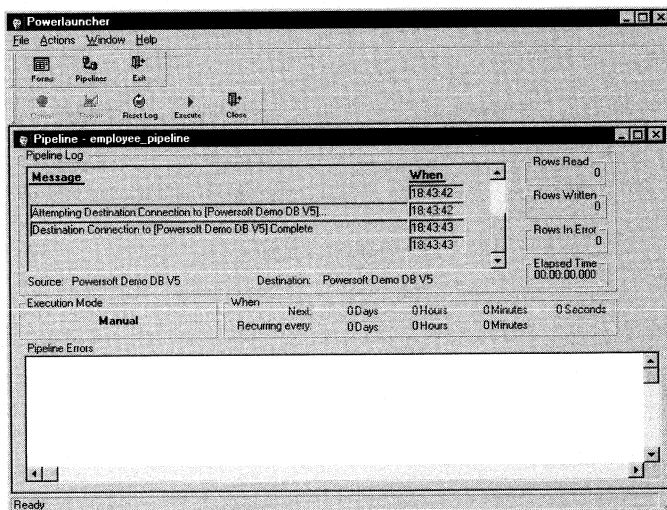
The Query Governor dialog box displays:



- 2 To specify the maximum number of rows to be retrieved and the maximum time for retrieval, select the appropriate checkbox, change the limit and click OK.

## Using a pipeline in an application

When you include a pipeline in an application and run it, the Pipeline window displays. As the window is displaying, messages are written to the Pipeline Log box to indicate connection to the source and destination databases, when these connections are occurring, and the elapsed time:



You can print the log or save the log to any report format. Users may need to send the log to you to use for debugging a pipeline. You can empty the log at any time by clicking the Reset Log button.

## Executing pipelines

A pipeline in an application executes when you execute it manually or you set a delayed execution mode to automatically execute the pipeline.

The execution mode (Manual by default) shows in the Execution Mode box. If you set a delayed execution modes, the time settings and a countdown clock display in the When box.

### Manual execution

You can manually execute a pipeline anytime, even if you've set a delayed execution mode.

- ❖ **To execute the pipeline manually:**
  - ◆ Click the Execute button.

**Delayed execution**

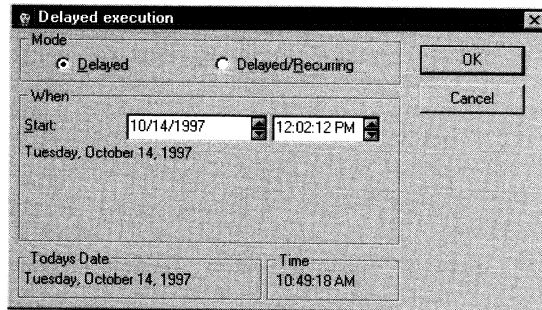
You can specify delayed automatic execution of the pipeline. This setting remains in effect until you close the Pipeline window.

❖ **To delay the execution of the pipeline:**

- 1 Select Actions>Delayed Execution from the menu bar.

The Delayed Execution dialog box displays.

- 2 Select the Delayed radio button if it's not already selected:



- 3 In the Start boxes, specify the execution date and time.

The day, month, day, and year for the date you specify displays under the Start boxes. You should verify that this date is the date you want because the date is based on the operating system's time zone settings. For example, depending on the time zone settings, a date of 10/12/97 could mean October 12th or December 10th.

- 4 Click OK.

The pipeline will execute at the specified date and time.

**Delayed/Recurring execution**

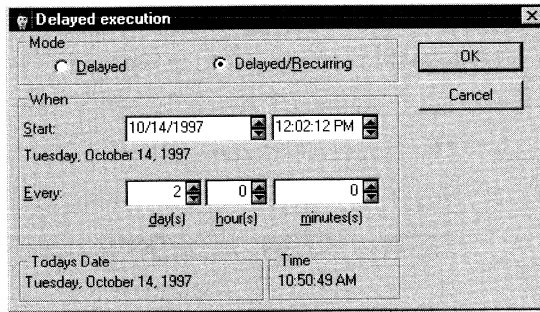
You can specify delayed and recurring automatic execution of the pipeline. This setting remains in effect until you close the Pipeline window.

❖ **To delay and repeat the execution of the pipeline at specified intervals:**

- 1 Select Actions>Delayed Execution from the menu bar.

The Delayed Execution dialog box displays.

- 2 Select the Delayed/Recurring radio button:



- 3 In the Start boxes, specify the execution date and time.

The day, month, day, and year for the date you specify displays under the Start boxes. You should verify that this date is the date you want because the date is based on the operating system's time zone settings. For example, depending on the time zone settings, a date of 10/12/97 could mean October 12th or December 10th.

- 4 In the Every boxes, specify the time delay for subsequent executions of the pipeline.
- 5 Click OK.

The pipeline will execute at the specified start time and then execute repeatedly.

If execution errors occur

If execution errors occur, the errors display in the Pipeline Errors box. You can print the errors or save the errors to any report format. You can sometimes correct errors in the Pipeline Errors box.

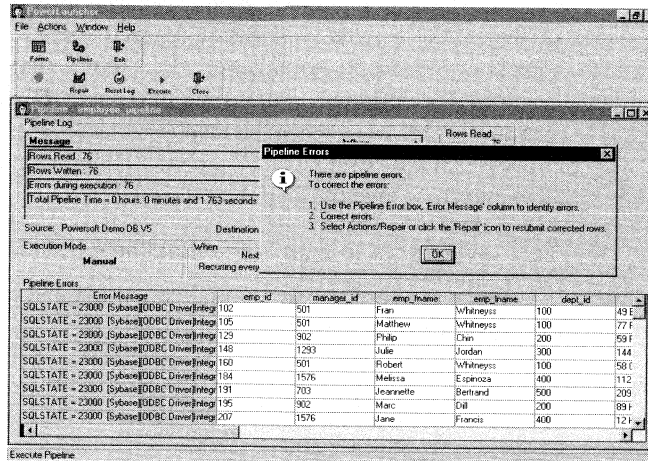
Users of the application usually should not repair pipeline execution errors. Users should contact you if execution errors occur and send the error log to you for debugging the pipeline. You should usually modify the pipeline object in InfoMaker to fix errors and redeploy the application.

At times, you may want users to fix execution errors by modifying the pipeline definition from the application. Advanced users of the application can modify the application's initialization file to allow them to modify the pipeline object's definition from the application. You can tell users how to do this if needed.

**FOR INFO** For information about adding a [Pipe] section to the initialization file, see "Modifying the pipeline object's definition" on page 633.

## Repairing execution errors

When a pipeline executes and has execution errors, the error messages display in the Pipeline Errors box and instructions for repairing the errors display:



### ❖ To repair pipeline errors:

- 1 In the Pipeline Errors box, look at the Error Message column to identify errors.
- 2 Change the data values for the appropriate columns in the error rows.  
You can extend the column borders to resize the column to the width needed to see error messages and column data.
- 3 Click the Repair button to execute the pipeline.

If errors have been corrected, the pipeline executes and the Pipeline Errors box clears.

## Modifying the pipeline object's definition

By default, you can execute a pipeline in an application, but you can't modify the pipeline object's definition.

But if you are deploying your application to advanced users, you may want to give users the ability to modify the pipeline object from the application. Allowing users to modify pipelines means that you don't have to modify the pipeline object in InfoMaker, recreate the application, and deploy it immediately. But since the pipeline object definition isn't saved when you modify a pipeline in an application, you should redeploy the application.

What you can allow users to modify

To modify a pipeline definition, either you or your users must add a new section to the application's initialization file. Users can change the type of pipeline operation and the Commit and Max Errors values. Depending on the quality of the network connection (particularly if users are connecting by telephone from laptops), lowering the Commit and Max Errors values could result in more efficient committing of rows to the database. For example, committing all rows when the database connection is through a 14.4 Kb/sec modem could take a long time and your phone connection could fail. So it may be better to change the Commit value to 10.

❖ **To enable users to modify the pipeline's definition:**

- 1 Open the application's initialization file and create a Pipe section by adding the following line:

```
[Pipe]
```

- 2 Include these line in the Pipe section:

```
[Pipe]
AllowTypeChange = 1
AllowRunTimeChange = 1
```

<b>This keyword</b>	<b>Allows the user to change the</b>
AllowTypeChange	Type of pipeline operation: Create, Refresh, Replace, Append, and Update
AllowRunTimeChange	Commit value, Max Errors value, and whether to pipe extended attributes

---

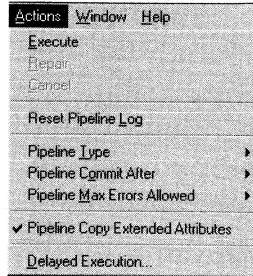
**Omitting lines in the [Pipe] section**

If you don't want users to change the pipeline type, you can omit the AllowTypeChange line in the [Pipe] section. You can't omit the AllowRunTimeChange line.

---

### 3 Save the initialization file.

Now when you select the Actions menu in a pipeline, you see new menu items that enable you to change the pipeline type, the Commit value, the Max Errors value, and whether the pipeline copies extended attributes:



#### Modifying the pipeline definition

After a user's application initialization file has been modified to include the [Pipe] section, the user can use items on the Actions menu to modify the pipeline definition.

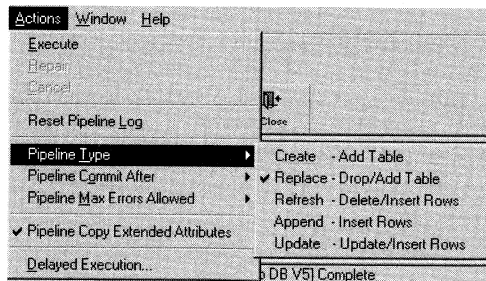
**FOR INFO** For complete information about pipeline types, the Commit value, and the Max Errors value, see Chapter 14, "Working with Data Pipelines".

#### ❖ To modify the pipeline type:

- 1 Select Actions>Pipeline Type from the menu bar.

A menu of pipeline types displays.

- 2 Select the pipeline type:

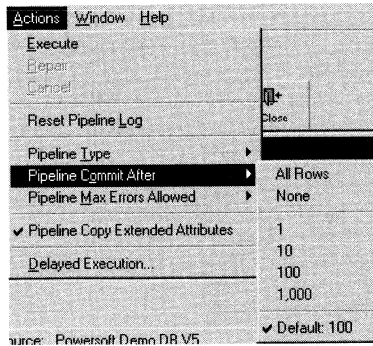


#### ❖ To modify the Commit value:

- 1 Select Actions>Pipeline Commit After from the menu bar.

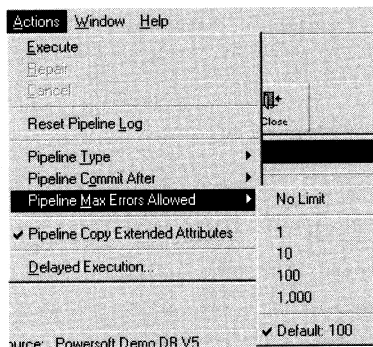
A menu of Commit values displays.

2 Select the Commit value:



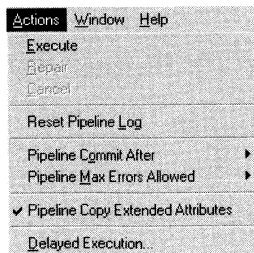
❖ To modify the Max Errors value:

- 1 Select Actions>Pipeline Max Errors Allowed from the menu bar.  
A menu of Max Errors values displays.
- 2 Select the Max Errors value:



❖ To specify whether or not to pipe extended attributes:

- ◆ Select or clear the Actions>Pipeline Copy Extended Attributes menu item:





# Deploying Your Application on Windows

## About this chapter

This chapter provides the information you need to deploy your application to your users' machines on the Windows platform.

## Contents

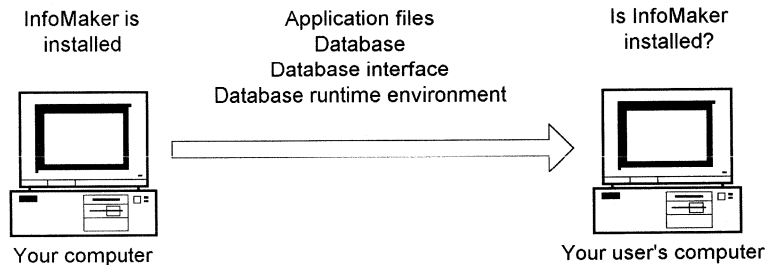
<b>Topic</b>	<b>Page</b>
About deploying applications	638
Deploying an application	639
Installing the Deployment Kit on a user's machine	640
Completing the deployment process	642
The deployed application	649
Deploying the SQL Anywhere Desktop Runtime System	655

## About deploying applications

Creating an executable version of an application means packaging an application that can run on its own, outside the InfoMaker environment. Users of your application do not need InfoMaker to run the application.

When you create an application, InfoMaker generates an executable file and an initialization file for your application. But to deploy an application, you must distribute files in addition to the executable file and the initialization file.

Exactly what you distribute depends on whether users have InfoMaker installed or not.



To set up a user's machine to run your application, Powersoft provides the Deployment Kit to install on the machine of users who do not have InfoMaker installed. When you install the Deployment Kit, all the deployment files and database interface files you need are installed in the right places. And if you are using a SQL Anywhere database, the SQL Anywhere files you need to run your application are also installed.

---

### Deploying on the Macintosh

When you create an application in InfoMaker for Windows, you can deploy the application only on Windows systems. If you want to deploy the application on Macintosh systems, you must create it again in InfoMaker for Macintosh using the same database and object library.

**FOR INFO** For information, see Chapter 19, "Deploying Your Application on the Macintosh" and Chapter 20, "Working in a Multiplatform Environment".

---

## Deploying an application

When you deploy an InfoMaker application, what you install on a user's computer depends on whether the user has InfoMaker installed.

If the user does not have InfoMaker installed, you need to install the following parts of the Deployment Kit:

What to install	Notes
Deployment DLLs	—
ODBC Drivers	If your database is an ODBC database, install the ODBC database driver
Native Database Drivers	If your database is a native database, install the native database driver
Additional Files	Check with your database vendor for information about additional files you need to deploy and the licensing issues

If the user has InfoMaker installed, you only need to install ODBC drivers from the Deployment Kit (if your database is an ODBC database).

### ❖ To deploy an application, do the following:

- 1 Create a folder for the application on the user's computer.
- 2 Copy the application executable and initialization files to the application folder.
- 3 Install the parts of the Deployment Kit required.
- 4 Distribute your data source.
- 5 Install the database runtime files on the user's computer.
- 6 Configure your ODBC drivers, system path, and registry files.

## Installing the Deployment Kit on a user's machine

---

### Running Setup from a 16-bit computer

Use the following procedure on a 32-bit computer to install the Deployment Kit on a 16-bit or 32-bit target computer. If you need to install from a 16-bit computer, navigate to the 16-bit DDDK directory on the CD and run the setup program from that directory.

---

### ❖ To install the Deployment Kit on a user's machine:

- 1 Exit any programs that are running.  
If you do not, the Setup program may not be able to copy some files to the appropriate directories.
- 2 Mount the CD-ROM and run the InfoMaker Setup program.
- 3 In the main Setup menu, click Deployment Kit.
- 4 Select the appropriate kit for the target machine (32-bit or 16-bit).
- 5 In the Deployment Kit dialog box, click Next.

---

### Back, Next, Cancel

Using Back and Next, you can step backward and forward through the installation process to modify specifications as necessary.

Using Cancel, you can halt the installation process. When you click Cancel, you can choose to either exit to the main Setup menu or resume the installation.

---

- 6 In the Choose Destination Location dialog box, click Next to accept the destination folder path shown.  
*or*  
Click Browse to select a different path, click OK, and click Next.
- 7 In the Confirm Path message box, click Yes to confirm the specified path.  
*or*  
Click No to return to the Choose Destination Location dialog box to specify another path.
- 8 In the Select Components dialog box, check the components you want to install.  
  
You can click Change to clear checkboxes for subcomponents you don't want to install.

- 9 In the Select Program Folder dialog box, specify the Program Folder to which program icons will be added and click Next.

Setup installs the components you selected. By default, Setup installs system DLLs in the appropriate system directory for your operating system, prompting you for an action if an update fails.

- 10 When prompted, select Yes to update the registry (on 32-bit systems) or AUTOEXEC.BAT (on 16-bit systems).

You must restart the computer before using the installed components.

- 11 In the Setup Complete dialog box, choose whether to restart the computer and click Finish to return to the main Setup menu.
- 12 Click Exit to exit the Setup program.

## Completing the deployment process

In addition to installing the Deployment Kit, you have a few other tasks to do before the deployment process is complete.

### Making the data source available

The Deployment Kit installs the database interface drivers you select. Your users also need access to the DBMS and to the database your application uses.

You need to:

- ◆ If necessary, install the DBMS runtime files in the application directory or in a directory on the system path.

If your application uses the SQL Anywhere Desktop Runtime System, you can install it using the setup program. For databases other than SQL Anywhere, follow the instructions and licensing rules specified by the vendor.

- ◆ Make sure each user has access to the database the application uses.

If your application uses a local database, install the database and any associated files, such as a log file, on the user's computer.

If your application uses a server database, make sure the user's computer is set up to access the database. This may be the task of a database administrator.

- ◆ If your application uses the ODBC interface, configure the ODBC database drivers and data sources, as described next.

### Configuring an ODBC driver

To use an ODBC data source in your application, the ODBC configuration must include:

- ◆ An entry for the DBMS driver in ODBCINST.INI
- ◆ An entry for the data source in ODBC.INI with instructions on how to start the database

Drivers in  
ODBCINST.INI

Installing the Deployment Kit installs ODBC drivers on the user's machine and modifies the registry (for Windows NT and Windows 95) or the user's ODBCINST.INI file (for Windows 3.x). ODBCINST.INI lists the ODBC drivers installed on the machine and the driver and setup DLLs for each driver.

For	ODBC driver information is in
Windows 3.x	ODBCINST.INI file
Windows 95 and Windows NT	Registry entry: HKEY_LOCAL_MACHINE\Software\ODBC\ODBCINST.INI

Installing the Deployment Kit installs all the deployment files into the directory you specify when you install. If you move the files to another location, you must modify the ODBCINST.INI file or registry entry. The following examples show typical locations for installed files.

**Windows 3.x** For example, if you install drivers for SQL Anywhere and dBASE, the ODBCINST.INI file has sections like the following:

```
[ODBC Drivers]
Sybase SQL Anywhere 5.5=Installed
PB Intersolv 2.12 dBASEFile=Installed

[Sybase SQL Anywhere 5.5]
Setup=C:\SQLANY50\WIN\WOD50W.DLL
Driver=C:\SQLANY50\WIN\WOD50W.DLL

[PB Intersolv 2.12 dBASEFile]
Setup=C:\WINDOWS\SYSTEM\PBdbf07.d11
Driver=C:\WINDOWS\SYSTEM\PBdbf07.d11
```

**Windows 95 or Windows NT** The registry entries are recorded in the ODBC drivers, Sybase SQL Anywhere, and PB INTERSOLV dBASEFile keys:

Key	Registry entry
ODBC Drivers	Sybase SQL Anywhere 5.5 Installed PB INTERSOLV 3.0 32-BIT dBASEFile (*.dbf) Installed
Sybase SQL Anywhere 5.5	Setup C:\SQLANY50\WIN32\WOD50T.DLL Driver C:\SQLANY50\WIN32\WOD50T.DLL
PB Intersolv 3.00 32-bit dBASEFile	Driver C:\WINDOWS\SYSTEM\PBDBF12.DLL Setup C:\WINDOWS\SYSTEM\PBDBF12.DLL

Data sources in  
ODBC.INI

To allow the user to connect to a particular data source, you must provide a definition for that data source. For Windows 3.x, the data source definition is in ODBC.INI; in Windows 95 and Windows NT, the entry is in the registry:

For	Data source entries are in
Windows 3.x	ODBC.INI file
Windows 95 and Windows NT	Registry entry: HKEY_CURRENT_USER\Software\ODBC\ODBC.INI

The Deployment Kit does not install data source definitions in ODBC.INI, so you need to do that for each ODBC database your application uses.

The data source must be listed in the section [ODBC Data Sources] with its associated DBMS and there must be a section whose name is the data source. The data source section specifies the connection parameters for the data source.

**Windows 3.x** For example, if your application uses a SQL Anywhere database named MyAppDB, ODBC.INI will include information like the following:

```
[ODBC Data Sources]
MyAppDB=Sybase SQL Anywhere 5.5

[MyAppDB]
Start=c:\SQLANY50\win\RTDSK50W.exe
Driver=:\SQLANY50\win\WOD50W.DLL
DatabaseName=MyAppDB
DatabaseFile=c:\PRODDIR\MyAppDB.db
```

**Windows 95 or Windows NT** The registry entries are recorded in the ODBC.INI key:

Key	Registry entry
ODBC Data Sources	MyAppDB Sybase SQL Anywhere 5.5
MyAppDB	Start C:\SQLANY50\WIN32\RTDSK50.EXE Driver C:\SQLANY50\WIN32\WOD50T.DLL DatabaseName MyAppDB DatabaseFile C:\PRODDIR\MyAppDB.DB



You might use one of the following ways to make the modifications to ODBC.INI:

- ◆ Use Microsoft's ODBC Administrator control panel, if the control panel is available on the user's system.
- ◆ Use a software distribution application that includes ODBC configuration instructions to do your installation.
- ◆ Look at your own ODBC.INI file that you have used during development and copy the relevant portions to the user's ODBC.INI. *This ad hoc method is not recommended, because there are no protections against incorrect modifications.* When you copy a section, make sure you modify paths to correspond with the user's setup.

**FOR INFO** For more information about the contents of the INI files for ODBC drivers and data sources, see *Connecting to Your Database*.

---

### **Database profiles**

*Connecting to Your Database* includes information about database profiles. Profiles, which are defined in IM.INI, provide the information necessary to connect to data sources from the *development* environment. Your users do not need database profiles.

---

## **Modifying the application's initialization file**

In your application's initialization file, remove your name from the line that reads:

UserID=*YourID*

An InfoMaker application's initialization file specifies the data source that is used by the reports and forms that were included in the application. When InfoMaker generated the initialization file, it included your User ID. When users run your application, they will be prompted to supply their user ID and password if:

- ◆ Your data source is not an ODBC data source
- ◆ Your data source is an ODBC data source and the user ID and password are not in the application's initialization file

## Deploying OCX controls

If your application uses OCX controls (OLE custom controls), you must:

- ◆ Deploy the OCX files with your application
- ◆ Make sure each OCX control is registered
- ◆ Make sure required files are in the target machine's system directory

### Registering OCX controls

The OCX controls provided with PowerBuilder are all self-registering—they register themselves when they are used. If you use an OCX control that is not self-registering, you need to register it manually on each user's machine. To find out whether a control is self-registering, see the documentation provided with the OCX control.

#### ❖ To register an OCX control:

- ◆ Run the `regsvr32.exe` (Windows 95 and NT) or `regsvr.exe` (Windows 3.x) command with the OCX filename as an argument.

For example, the following commands register the 32-bit and 16-bit versions of an OCX control called ABCDE:

```
regsvr32.exe abcde32.ocx  
regsvr.exe abcde16.ocx
```

---

### Troubleshooting OCX registration

If you have trouble registering an OCX control, make sure that the same versions of the following DLLs are available on both the development and target machines:

- ◆ **Windows 3.x** OC25.DLL
- ◆ **Windows 95 and NT** MFC40.DLL, MFC42.DLL, OLEPRO32.DLL, MSVCRT40.DLL, STDOLE2.TLB, CTL3D32.DLL, WININET.DLL

If the versions do not match in either size or date, copy them from the development machine to the target machine and register OC25.DLL (Windows 3.x) or MFC40.DLL and OLEPRO32.DLL (Windows 95 and NT) on the target machine. There are different versions of CTL3D32.DLL for Windows 95 and NT.

### Required files

Depending on the development and deployment platforms and the OCX controls you are deploying, you may need to copy additional DLLs or license files to the WINDOWS or WINNT system directories on the target machine.

**FOR INFO** For more information, see "OLE system files" on page 653 and the documentation provided with the OCX control. You may also want to look in the Powersoft Tools Technical Information Library on the Web.

## Starting the deployed application

Your users can run your application the same way they run other Windows applications. For example, they can double-click the executable file in Explorer or File Manager, or they can create an application shortcut on the desktop and double-click the shortcut.

If users create a shortcut, the Target textbox on the Shortcut properties page should specify the path to the executable, and the Start In textbox should specify the location of the Powersoft deployment DLLs.

**FOR INFO** For information about modifying an application shortcut's properties and running an application, see Chapter 17, "Working with Applications".

## The deployed application

In this section you will find lists of the files a user needs for running an InfoMaker application. *If you select one of the Deployment Kit options in the Setup program, the required files are installed for you.* The lists in this section tell you what files get installed so that you can do an installation manually or so that you can specify file lists for a Setup program.

The lists do not include the application files (executable and initialization files), the database files, and the files that provide the database's runtime environment.

### Localization kits

InfoMaker provides localized deployment kits (DDDK) as an installation option. They are available in French, German, Italian, Spanish, Dutch, Danish, Norwegian, and Swedish.

You can install the DDDK in the development environment or on the user's machine. If you install the DDDK on the development machine, you can use it for testing purposes.

The localized deployment kits handle language-specific data at execution time.

### Filenames

DLLs have different names for different deployment environments so that you can tell at a glance whether you are deploying the correct versions of files. In the following tables, names are shown like this: PBVM60x.DLL. The x is replaced with a letter for deployment platforms other than the ANSI version of Windows 95 and Windows NT.

For example, if your application includes reports, you will deploy PBDWE60.DLL on 32-bit Windows platforms and PBDWE60W.DLL on 16-bit Windows platforms.

The tables indicate which feature or DBMS the file is required to support and whether it is available on Windows 3.1 or Windows 95 and Windows NT. For example, PBVM60x.DLL is required for all 32-bit deployed applications. You need not install other files unless your application uses the features indicated.

### Powersoft files

The Powersoft deployment DLLs belong in the application directory or in a directory on the system path.

---

### Deployment Kits the same for PowerBuilder and InfoMaker

The Deployment Kits are the same for PowerBuilder and InfoMaker. Some of the files listed in the following table support features that are provided only with PowerBuilder.

---

Required for	Name	Windows 3.x	Windows 95 and NT
All	PBVM60x.DLL	X	X
Reports	PBDWE60x.DLL	X	X
Rich Text	PBRTC60x.DLL	X	X
OLE	PBOUI60W.DLL	X	—
OLE automation servers	PBAEN60.tlb	—	X
All	PBBGR60W.DLL	X	—
All	PBTYP60W.DLL	X	—
Visual Basic controls	PBVBX60W.DLL	X	—
DataWindow plug-in	NPDWE60x.DLL	X	X
Standard PowerBuilder window plug-in	NPPBA60x.DLL	X	X
Secure PowerBuilder window plug-in	NPPBS60x.DLL	X	X
Standard PowerBuilder window ActiveX	PBRX60x.OCX	—	X
Secure PowerBuilder window ActiveX	PBRXS60x.OCX	—	X
PowerBuilder Synchronizer executable	PBSYNCx.EXE	X	X

Native database interfaces

The files for the native Powersoft database interfaces your application uses belong in the application directory or a directory on the system path:

DBMS	Name	Windows 3.x	Windows 95 and NT
IBM databases	PBIBM60W.DLL	X	—
INFORMIX I-Net 5	PBIN560x.DLL	X	X
INFORMIX I-Net 7	PBIN760x.DLL	—	X
Microsoft SQL Server 4.x DB-Lib	PBSYB60x.DLL	X	X
	PBDBL60x.DLL	X	X
Microsoft SQL Server 6.0	PBMSS60x.DLL	X	X

DBMS	Name	Windows 3.x	Windows 95 and NT
Powersoft ODBC Interface	PBODB60x.DLL	X	X
	PBODB60x.INI	X	X
Oracle Version 7.1	PBO7160x.DLL	X	X
Oracle Version 7.2	PBO7260x.DLL	X	X
Oracle Version 7.3	PBO7360x.DLL	X	X
Sybase InformationCONNECT DB2 Gateway	PBMDI60x.DLL	X	X
	PBDBL60x.DLL	X	X
Sybase Net-Gateway for DB2	PBNET60x.DLL	X	X
	PBDBL60x.DLL	X	X
Sybase SQL Server 4.x DB-Lib	PBSYT60x.DLL	—	X
	PBDBT60x.DLL	—	X
Sybase SQL Server 10.x and 11.x CT-Lib	PBSYC60x.DLL	X	X

### ODBC and system files

If your application uses ODBC drivers, each user's machine needs four types of files:

- ◆ **Powersoft database interface files** PBODB60x.DLL and PBODB60x.INI belong in the application directory or a directory on the system path.
- ◆ **ODBC initialization files** For more information about ODBC.INI and ODBCINST.INI, see "Configuring an ODBC driver" on page 642.
- ◆ **Microsoft ODBC driver and DLLs** These files belong in the WINDOWS\SYSTEM or WINNT\SYSTEM directory:

Windows 3.x	Windows 95 and NT
CPN16UT.DLL	DS16GT.DLL
ODBC16UT.DLL	DS32GT.DLL
ODBC.DLL	MSVCRT40.DLL
ODBCCURS.DLL	ODBC32.DLL
ODBCINT.DLL	ODBC16GT.DLL
ODBCINST.HLP	ODBC32GT.DLL

Windows 3.x	Windows 95 and NT
ODBCADM.EXE	ODBCCP32.CPL
	ODBCCP32.DLL
	ODBCCR32.DLL
	ODBCINT.DLL
	ODBCINST.CNT
	ODBCINST.HLP
	ODBCTRAC.DLL
	ODBCAD32.EXE

- ◆ **ODBC database drivers and supporting files** These files belong in the WINDOWS\SYSTEM or WINNT\SYSTEM directory.

---

**Filenames may change**

Many of the files listed here are provided by vendors other than Powersoft. More recent files with different names may be substituted at any time.

---

ODBC Drivers	Windows 3.x	Windows 95 and NT
All INTERSOLV drivers	PBBAS07.DLL	PBBAS12.DLL
	PBFLT07.DLL	PBFLT12.DLL
	PBUTL07.DLL	PBUTL12.DLL
	PBDRV07.CNT	PBDRV12.CNT
	PBDRV07.HLP	PBDRV12.HLP
	QETRN07.DLL	PBTRN12.DLL
	QEPB.LIC	IVPB.LIC
INTERSOVL Btrieve	PBBTR07.DLL	PBBTR12.DLL
	WBTRCALL.DLL	PBBTR12.HLP
	PBBTR07.HLP	
INTERSOVL dBASE	PBDBF07.DLL	PBDBF12.DLL
	PBDBF07.HLP	PBDBF12.HLP
INTERSOVL DB2	PBDB207.DLL	PBDB212.DLL
	PBDB207.DLL	PBDB212.DLL



ODBC Drivers	Windows 3.x	Windows 95 and NT
INTERSOLV Excel 4	PBXLS07.DLL PBXLS07.HLP	—
INTERSOLV Excel Workbook	PBXLS507.DLL PBXLS507.HLP	PBXLS512.DLL PBXLS512.HLP
INTERSOLV Scalable SQL	PBXQL07.DLL WDDLVCVS.DLL WXQLCALL.DLL PBXQL07.HLP	—
INTERSOLV Paradox 4	PBPDX07.DLL PBPDX07.HLP	—
INTERSOLV Paradox 5	PBIDP07.DLL PBIDP07.HLP	PBIDP12.DLL PBIDP12.HLP
INTERSOLV Text	PBTXT07.DLL PBTXT07.HLP	PBTXT12.DLL PBTXT12.HLP
Sybase SQL Anywhere 5.5	WOD50W.DLL WL50EN.DLL WTR50W.DLL WODBC.HLP	WOD50T.DLL WL50ENT.DLL WTR50T.DLL WODBC.HLP

## OLE system files

**Windows 3.x** Most users will have the required OLE DLLs on their systems because they have already installed an OLE server application. If the user's system does not have the OLE DLLs, copy them to the WINDOWS\SYSTEM directory. To allow users to access OLE 1.0 applications, you must also merge the file OLE2.REG into the system registry using Microsoft's REGEDIT utility.

COMPOBJ.DLL  
CTL3DV2.DLL  
OLE2.DLL  
OLE2CONV.DLL  
OLE2DISP.DLL  
OLE2NLS.DLL  
OLE2PROX.DLL  
STORAGE.DLL  
TYPELIB.DLL

**Windows 95 and Windows NT** For Windows 95 and Windows NT, the OLE DLLs are part of the system and do not need to be installed separately.

## Deploying the SQL Anywhere Desktop Runtime System

If your InfoMaker application has reports and forms that access a SQL Anywhere database, you need to deploy the SQL Anywhere DBMS as well as SQL Anywhere's ODBC database drivers described in the preceding section. You can deploy the SQL Anywhere Desktop Runtime System without incurring additional license fees. The Runtime System allows the user to retrieve and modify data in the database, but does not allow modifications to the database schema. It does not support transaction logs, stored procedures, or triggers.

You can install the SQL Anywhere Desktop Runtime System using InfoMaker's Deployment Kit. See "Installing the Deployment Kit on a user's machine" on page 640.

---

### Other SQL Anywhere products

InfoMaker includes SQL Anywhere for use during the development process. However, this product cannot be deployed royalty-free to your users.

If your application requires the data definition language (DDL), a transaction log, stored procedures, or triggers, see your Powersoft sales representative.

---

In addition to deploying the runtime system, you need to do the following for your users:

- ◆ Make your application's data source available
- ◆ Configure ODBC to access the data source

These tasks are described next.

## Making the data source available

- ❖ **To make a SQL Anywhere database available to a user:**
  - ◆ Copy the database and its log file, if necessary, to the user's computer.

Although the Runtime System does not use the log file, it will access it the first time you start the database to make sure the database is in a consistent state. Then it will rename the log file and run without using the log. The Runtime System does not delete the log file, because it might be needed later (although this is unlikely). You do not need the log file if there were no errors the last time you closed the database.

## Configuring SQL Anywhere's ODBC driver and data sources

To use a SQL Anywhere database in your application, the ODBC configuration must include:

- ◆ An entry for SQL Anywhere in ODBCINST.INI
- ◆ An entry for the data source in ODBC.INI with information about how to start the database

Driver in  
ODBCINST.INI

When you use the Deployment Kit to install the SQL Anywhere Desktop Runtime System on a user's system, the Setup program adds the appropriate information to ODBCINST.INI.

**FOR INFO** For examples of resulting files, see "Configuring an ODBC driver" on page 642.

Data sources in  
ODBC.INI

Setup does not create entries for individual databases in ODBC.INI. There are several ways for you to do this.

If you use the ODBC Administrator control panel, check the information about SQL Anywhere in *Connecting to Your Database*. Even though this information covers connecting to the database within the development environment, the description of the fields in the SQL Anywhere ODBC Configuration dialog box is applicable.

For each data source, you specify how ODBC will connect to the DBMS. To connect to the Runtime System, you specify the Runtime System database engine as the Start Command in the ODBC Administrator control panel.

### ❖ To configure ODBC to start the Runtime System engine:

- ◆ Follow the procedure for specifying startup options in *Connecting to Your Database*. For the Start Command, specify one of the runtime engines:

On this operating system	Specify this engine
Windows 3.x	rtdsk50w

On this operating system	Specify this engine
Windows 95 or Windows NT	rtask50

FOR INFO For information on startup options, see the description of DBSTART in the *SQL Anywhere User's Guide* or online Help.

## The deployed SQL Anywhere runtime files

If you use the Deployment Kit to install the SQL Anywhere Desktop Runtime System, the files listed in this section are installed for you.

The deployed  
Runtime System

These files belong in a directory on the system path:

Description	Windows 3.x	Windows 95 and NT
Runtime database engine	RTDSK50W.EXE	RTDSK50.EXE
Database interface DLL	DBL50W.DLL	DBL50T.DLL
English strings for the database interface	WL50EN.DLL	WL50ENT.DLL
ODBC driver	WOD50W.DLL	WOD50T.DLL
Help file for configuring ODBC	WODBC.HLP	WODBC.HLP
ODBC translation DLL	WTR50W.DLL	WTR50T.DLL
Back up a database	DBBACKW.EXE	DBBACKUP.EXE
Erase a database	DBERASEW.EXE	DBERASE.EXE
Display information about a database	DBINFOW.EXE	DBINFO.EXE
Stop the database engine or a client	DBSTOPW.EXE	DBSTOP.EXE
Unload a database	DBUNLOAW.EXE	DBUNLOAD.EXE
Validate the integrity of a database	DBVALIDW.EXE	DBVALID.EXE
Create a database write file	DBWRITEW.EXE	DBWRITE.EXE
Database tools utility DLL	DBTL50W.DLL	DBTL50T.DLL

---

**Reminder**

Put the directory for the SQL Anywhere Desktop Runtime System on the system path.

---

# Deploying Your Application on the Macintosh

## About this chapter

This chapter provides the information you need to deploy your application to your users' machines on the Macintosh platform.

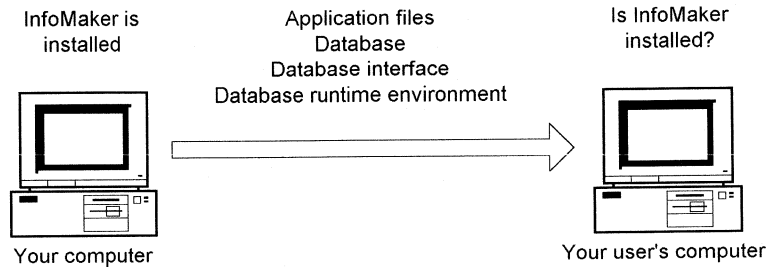
## Contents

<b>Topic</b>	<b>Page</b>
About deploying applications	660
Basic steps for deploying an application	661
Installing the Deployment Kit on a user's machine	662
Completing the deployment process	663
Installing application files	665
The deployed application	666

## About deploying applications

Creating an executable version of an application means packaging an application that can run on its own, outside the InfoMaker environment. Users of your application do not need InfoMaker to run the application.

When you create an application, InfoMaker generates an executable file (a program file) and an initialization file (a Preferences file) for your application. But to deploy an application, you must distribute files in addition to the executable file and the initialization file. Exactly what you distribute depends on whether users have InfoMaker installed.



To set up a user's machine to run your application, Powersoft provides the Deployment Kit to install on the machine of users who do not have InfoMaker installed. When you install the Deployment Kit, all the deployment files and database interface files you need are installed in the right places. And if you are using a SQL Anywhere database, the SQL Anywhere files you need to run your application are also installed.

---

### Deploying on Windows

When you create an application in InfoMaker for Macintosh, you can deploy the application only on Macintosh systems. If you want to deploy the application on Windows systems, you must create it again in InfoMaker for Windows using the same database and object library.

FOR INFO For information, see Chapter 18, "Deploying Your Application on Windows" and Chapter 20, "Working in a Multiplatform Environment".

---



## Basic steps for deploying an application

This checklist provides an overview of the process of deploying your InfoMaker application on the Macintosh. These tasks vary depending on whether the user has InfoMaker installed. These actions must be performed on *each user's* Macintosh.

### Deployment checklist

InfoMaker installed	InfoMaker not installed	Action
X	X	Create an application folder
	X	Create a Powersoft 6.0 Folder in System Folder:Extensions
X	X	Prepare users' machines by installing application and environmental components FOR INFO For instructions, see "Installing the Deployment Kit on a user's machine" next
X	X	Make the data source available by installing DBMS runtime files and the database FOR INFO For instructions, see "Making the data source available" on page 663
	(X)	If you installed an ODBC interface, a SQL Anywhere driver, or another Apple ODBC 2.0-compliant driver, configure the ODBC driver FOR INFO For instructions, see "Configuring an ODBC driver" on page 663
X	X	Install the application files FOR INFO For instructions, see "Installing application files" on page 665

## **Installing the Deployment Kit on a user's machine**

❖ **To install the Deployment Kit on a user's machine:**

- 1 Exit any programs that are running.
- 2 Insert the CD-ROM in the CD-ROM drive.
- 3 Double-click Install Deployment Kit.
- 4 Select an installation option:
  - ◆ **Easy Install** Installs all the components of the Deployment Kit.
  - ◆ **Custom Install** Allows you to select the options you want to install. Some of the options have additional choices. Click the triangle to display them. For information about an option, click the "I" button to its right.
- 5 Click OK.
- 6 The installer will copy the appropriate files to the appropriate location in the System folder.

## Completing the deployment process

In addition to installing the Deployment Kit, you have a few other tasks to do before the deployment process is complete.

### Making the data source available

The Deployment Kit installs the database interface drivers you select. Your users also need access to the DBMS and to the database your application uses.

You need to:

- ◆ If necessary, install the DBMS runtime files.  
Follow the instructions and licensing rules specified by the vendor. If you are using a SQL Anywhere database, your runtime files will be available to each user. You need take no action.
- ◆ Make sure each user has access to the database the application uses.  
If your application uses a local database, install the database and any associated files, such as a log file, on the user's computer.  
If your application uses a server database, make sure the user's computer is set up to access the database. This may be the task of a database administrator.
- ◆ If your application uses the ODBC interface, configure the ODBC database drivers and data sources, as described next.

### Configuring an ODBC driver

Data sources in  
ODBC Preferences

To allow the user to connect to a particular data source, you must provide a definition for that data source in ODBC Preferences. The Deployment Kit does not install data source definitions in ODBC Preferences, so you need to do that for each ODBC database your application uses.

To configure an ODBC driver, use the ODBC Setup dialog box for your driver.

---

**Do not edit the ODBC Preferences file directly**

You could use a text editor to look at the ODBC Preferences that you used during development and copy the relevant portions to the user's ODBC Preferences. *This ad hoc method is not recommended, because there are no protections against incorrect modifications.* When you copy a section, you must make sure to modify paths so that they correspond with the user's setup.

---

FOR INFO For more information about ODBC configuration for drivers and data sources, see *Connecting to Your Database*.

---

**Database profiles**

*Connecting to Your Database* includes information about database profiles. Profiles are defined in InfoMaker Preferences in System Folder:Preferences:Powersoft 6.0 Preferences. They provide the information necessary to connect to data sources from the development environment. Your users do not need database profiles.

---

## Installing application files

You need to make sure your application files are installed on each user's computer, in the folder where your application expects them to be. First you need to modify your application's initialization file. Then you need to install the files and test the application.

Modifying the application's initialization file

In your application's initialization file, remove your name from the line that reads:

```
UserID=YourID
```

An InfoMaker application's initialization file specifies the data source that is used by the reports and forms that were included in the application. When InfoMaker generated the initialization file, it included your user ID. When users run your application, they will be prompted to supply their user ID and password if:

- ◆ Your data source is not an ODBC data source
- ◆ Your data source is an ODBC data source and the user ID and password are not in the application's initialization file

Installing the application and initialization files

Copy the application file and the application's initialization file to the application folder you created on each user's computer.

Testing the deployed application

You should test the deployed application on one user's Macintosh to make sure everything performs as expected. You can start the application by double-clicking the application icon in the application folder, or by creating an alias on the desktop to point to the application and double-clicking the alias.

## The deployed application

In this section you will find lists of the files a user needs for running an InfoMaker application. If you select one of the Deployment Kit options in the Installer, the required files are installed for you. By default, the files are installed in the same locations as they are when you install InfoMaker. You can also install files in the same directory as the application or in the system path.

The lists and tables in this section tell you what files get installed by the Deployment Kit and where the default location is so that you can do an installation manually or so that you can specify file lists for your own Installer. The files are required on both the Power Macintosh and 68K Macintosh unless otherwise noted.

### InfoMaker files

The InfoMaker shared libraries for deployment are installed in System Folder:Extensions:Powersoft 6.0.

---

#### Deployment Kits the same for PowerBuilder and InfoMaker

The Deployment Kits are the same for PowerBuilder and InfoMaker. Some of the files listed in the following table support Internet features that are provided only with PowerBuilder.

---

Required for	Name	68K Macintosh	Power Macintosh
All	PowerBuilder 6.0 VM Library	X	X
All	Macintosh Library	X	X
DataWindows	DataWindow Library	X	X
Rich Text	RichText Library	X	X
DataWindow plug-in	DataWindow Plugin	—	X
Standard PowerBuilder window plug-in	PowerBuilder Window Plugin	—	X
Secure PowerBuilder window plug-in	PowerBuilder Secure Window	—	X

### Preferences files

The Deployment Kit installs Preferences files for ODBC, Powersoft labels, and Powersoft font mappings in System Folder:Preferences.

## Extensions for 68K Macintosh

On the 68K Macintosh platform, the following files are installed in System Folder:Extensions when you install either the Powersoft deployment libraries or Apple System Software:

- ◆ CFM-68K Runtime Enabler
- ◆ Thread manager
- ◆ ObjectSupportLib
- ◆ AppleScriptLib

## Powersoft database interfaces

The files for the Powersoft database interfaces your application uses are installed in System Folder:Extensions:Powersoft 6.0:

Name	68K Macintosh	Power Macintosh
ODBC Driver	X	X
OR7 Oracle 7.0 Driver	X	X
O71 Oracle 7.1 Driver	X	X
073 Oracle 7.3 Driver	—	X
SYB SQL Server 4.x Driver	X	X
SYC Sybase System 10/11 Driver	X	X

## ODBC and system files

If your application uses ODBC drivers, you need the files listed in the following table. You also need to install the Powersoft ODBC Driver (listed under Powersoft database interfaces) and, if you want to access a DBMS other than SQL Anywhere, an ODBC-compliant driver for the DBMS you want to access.

**FOR INFO** For more information about ODBC support and obtaining ODBC drivers, see *Connecting to Your Database*.

The following files are installed by the Deployment Kit when you select Apple System Software and then ODBC. The Power Macintosh versions of these filenames end with PPC:

Filename	Location
ODBC Preferences [PPC]	System Folder: Preferences
ODBC Driver Manager [PPC]	System Folder: Extensions
ODBC Configuration Manager [PPC]	System Folder: Extensions
ODBC Cursor Library [PPC]	System Folder: Extensions

<b>Filename</b>	<b>Location</b>
ODBC Setup [PPC]	System Folder: Control Panels

---

### **ODBC Preferences**

If an ODBC Preferences file already exists on the user's machine, do not replace it with a new one. Data sources already defined for the user will be lost. You can use the ODBC Setup dialog box for your driver to add your application's data sources.

FOR INFO See "Configuring an ODBC driver" on page 663.

---

### **SQL Anywhere and deployment**

SQL Anywhere is part of the InfoMaker package for use in the development environment. If you want to deploy SQL Anywhere with your InfoMaker application, each user must have his or her own copy of SQL Anywhere. See your Powersoft sales representative for information.

---



# Working in a Multiplatform Environment

## About this chapter

This chapter describes files you can share in a multiplatform InfoMaker environment and things you will want to consider as you work with others in that environment.

## Contents

<b>Topic</b>	<b>Page</b>
Your working environment	670
Sharing files in a multiplatform environment	671
Applications for Macintosh and Windows	676
Saving text files—what line endings to use	677
Designing multiplatform reports and forms	678
Platform support for InfoMaker features	684

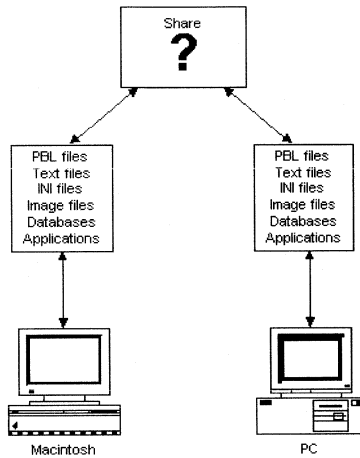
## Your working environment

### Single platform

If you work in an environment that has only Macintosh or only Windows systems, you and your coworkers running InfoMaker for Macintosh can share seamlessly what you create. In this case, you may not need to read this chapter.

### Macintosh and Windows

If you work in an environment in which InfoMaker is running on both Macintosh and Windows, you and your coworkers can often share what you create on either platform. However, you need to know what the guidelines are so you can consider them as you create objects and other items you will share:



## Sharing files in a multiplatform environment

In a multiplatform environment, you and your coworkers who are using InfoMaker on different platforms can share many files, with only a few restrictions.

### Files you can share

Among the files you can share are:

- ◆ Object libraries (PBL files)
- ◆ Text files of several types
- ◆ Initialization files
- ◆ Image files of several types
- ◆ SQL Anywhere databases (DB and LOG files)

### General guidelines

**Accessing files on shared drives** You and your coworkers using InfoMaker on different platforms can all access files that are stored on shared drives. If the InfoMaker versions are the same, you don't need to move the files to a particular platform or perform any conversions to use them.

**Moving files** You can move InfoMaker files and resources to any platform. You don't need to convert text files. And depending on the InfoMaker version on each platform, you may not need to regenerate libraries.

**Filenames** In a multiplatform environment, it's easiest to use the Windows convention for filenames—one to eight characters followed by a period and a 3-character extension. If you use a long filename on Macintosh and move a file to Windows, the filename will be truncated on the Windows system.

## Object libraries

In InfoMaker, you create reports, forms, queries, and data pipelines that are kept in both source and binary form in PowerBuilder libraries (PBLs). You can include reports and forms in an InfoMaker application.

### PBLs on shared drives

You and your coworkers using InfoMaker on different platforms can access objects in PBLs stored on shared drives. You and your coworkers can also create objects, because a PBL can contain objects created in both InfoMaker for Macintosh and InfoMaker for Windows. You don't need to move the files to a particular platform or perform any conversions to use them.

Copying and moving PBLs

You can copy and move PBLs to both Windows and Macintosh platforms, with these restrictions:

- ◆ If the InfoMaker versions are the same (for example, both are 6.0) on both the platform that created the PBL and the platform that uses the PBL, you can use the PBL with no change, sharing the same file on both platforms.
- ◆ If the InfoMaker versions are different (some are Version 6.0 and others are 5.0), you cannot share the PBL.

When you copy or move a PBL from Windows to the Macintosh, its file type and creator are not set and it will not have the correct icon in the Finder. The first time you open the library in InfoMaker, InfoMaker automatically sets this information so that the icon is correct.

Problems with sharing some objects

**OLE blob columns** An InfoMaker for Windows report can contain an OLE blob column. OLE is not supported in InfoMaker for Macintosh. If you open such a report in InfoMaker for Macintosh, you can't save changes to the report unless you delete the OLE blob column.

**Windows metafiles** InfoMaker for Macintosh doesn't support Windows metafiles (WMF). If you open a report created in InfoMaker for Windows that uses a metafile for a picture, the picture does not display. If you open a form created in InfoMaker for Windows that uses a metafile for a picture or picture button, the picture does not display.

**Macintosh PICT files** InfoMaker for Windows doesn't support Macintosh PICT files. If you open a report created in InfoMaker for Macintosh that uses a PICT file for a picture, the picture does not display.

Performance

Reports and forms run faster in InfoMaker and in an InfoMaker application if the reports and forms are created (or at least opened and resaved) on the same platform they will run on.

## Text files

In general, you don't need to convert files that you have copied or moved to another platform.

Macintosh and Windows systems use different line endings (also called line terminators) for text files. But in most cases, InfoMaker for Macintosh and InfoMaker for Windows recognize all line endings as valid and can read all text files correctly.

## Initialization files

The reading and writing of values in initialization files differs for InfoMaker for Macintosh and InfoMaker for Windows.

InfoMaker users do not usually access initialization files they create as PowerBuilder users do. But you could access an initialization file using the ProfileInt function and ProfileString function in a computed field in a form or report to get a value or string in an initialization file. The following information is provided in the event you do this.

Sharing information  
in an initialization file

**On Windows** InfoMaker for Windows can only access an initialization file that uses native Windows line endings. To share the information in an initialization file created on a Macintosh platform, you must make a copy of the initialization file and convert it using an appropriate utility.

**On Macintosh** InfoMaker for Macintosh can access initialization files from both Windows and Macintosh. If the contents of an initialization file are applicable on both platforms, you can access a Windows initialization file stored on a shared drive. However, if you write to the initialization file, the file will have a mix of line endings and become inaccessible by a coworker using InfoMaker for Windows. Generally, initialization files shared in this way are read-only, so this should not be a problem.

Don't share the  
InfoMaker  
initialization file itself

The InfoMaker initialization file contains configuration information for many of InfoMaker's painters. Much of this information is in the form of filenames and paths and is not meaningful when InfoMaker is run on the Macintosh. Therefore, you and coworkers using InfoMaker on different platforms need your own InfoMaker initialization files and cannot share them from a network drive.

Using profiles  
created on a different  
platform

Database profiles, which are stored in the InfoMaker initialization file, are usable on any platform. If you have created database profiles on a Macintosh platform, you can transfer them to a Windows platform. You do this by converting the line endings in the InfoMaker initialization file from the Macintosh platform and then cutting and pasting the database profile sections into the InfoMaker initialization file being used on the Windows platform. You can use a text conversion tool or InfoMaker's file editor to open the file and save it in the native format.

**FOR INFO** For information about how to save files in native format, see "Saving text files—what line endings to use" on page 677.

Sharing profiles

**On Windows** The procedures for allowing several users to read profiles from one initialization file (described in *Connecting to Your Database*) only applies when the initialization file has Windows line endings.

**On Macintosh** You can use the procedure for allowing several users to read profiles from one initialization file (described in *Connecting to Your Database*) to access an initialization file created on any platform. Because this file is read-only for the local user, there will be no conflicting line endings added by writing to this shared file.

## Image files

InfoMaker supports most Windows file types for images on the Macintosh:

Image file	InfoMaker for Windows	InfoMaker for Macintosh
16-color and 256-color bitmaps (BMP)	x	x
Run-length-encoded (RLE)	x	x
Icons (ICO)	x	x
Pointers or cursors (CUR)	x	x
Windows metafiles (WMF)	x	—
Macintosh bitmap files (PICT)	—	x

Where to create

Since the tools for creating most of these types of files are not common on the Macintosh, you may want to create images in Windows and copy them to the Macintosh.

WMF format

Window metafiles (WMF) format is only supported by InfoMaker on Windows. If you open a report or form created in InfoMaker for Windows that uses a Windows metafile for a picture, the picture does not display in InfoMaker for Macintosh.

PICT format

Macintosh bitmap (PICT) files are only supported by InfoMaker for Macintosh. If you open a report or form created in InfoMaker for Macintosh that uses a Macintosh bitmap for a picture, the picture does not display in InfoMaker for Windows.

## SQL Anywhere databases

You can move SQL Anywhere databases between Windows and Macintosh and access them from either platform. SQL Anywhere databases are platform independent.

After you move a SQL Anywhere database from Windows to the Macintosh, drag the Windows DB file and drop it on the SQL Anywhere icon. SQL Anywhere converts the file and changes the Windows icon to the appropriate Macintosh SQL Anywhere icon:



When you copy or move a SQL Anywhere database from Windows to Macintosh or from Macintosh to Windows, you should copy or move its LOG file with it.

**FOR INFO** For complete information about accessing databases in InfoMaker for Macintosh, see *Connecting to Your Database*.

## Applications for Macintosh and Windows

Although object libraries can be shared, InfoMaker applications can't be shared directly. Because an InfoMaker application contains platform-specific information, it must be created specifically for each platform and can be deployed only on the platform on which it was created.

### ❖ To use an application on both Macintosh and Windows:

- 1 Be careful placing objects in reports and forms and handling fonts and text so that the resulting reports and forms look correct with nothing cut off or overlapping.

FOR INFO For more information, see "Designing multiplatform reports and forms" on page 678.

- 2 If InfoMaker isn't installed on the other platform, install it.
- 3 If you are using SQL Anywhere, copy the database to the other platform.
- 4 Copy the PBL that contains the reports and forms you want to include in the application to the other platform.
- 5 (Optional) Use PowerBuilder to regenerate the objects in the PBL. Doing this before you create the application on the other platform improves performance.
- 6 Create the application using the Environment painter on the other platform.



## Saving text files—what line endings to use

When reading a text file, InfoMaker detects the characters used for line endings and reads the file correctly. When writing a text file, InfoMaker writes the file using line endings appropriate for the current platform unless you specify otherwise by setting a variable for the file editor in the InfoMaker Preferences file.

### ❖ To specify line endings in InfoMaker:

- ◆ In the InfoMaker Preferences file, set the `EditorExportFileType` preference variable:

To specify line endings for this platform	Set EditorExportFileType to
Current platform (default)	0
Windows (CR/LF)	1
Macintosh (CR only)	2

For example, the [PB] section of the InfoMaker initialization file might include a line like this:

```
[PB]
EditorExportFileType=0
```

## Designing multiplatform reports and forms

The differences between the Macintosh and Windows platforms affect the layout of objects and fonts in reports and forms. The differences in screen resolutions and pixel size on Macintosh and Windows also affect the look of the same reports and forms on Macintosh and Windows.

If you will be using the same reports and forms on both Macintosh and Windows platforms, you should be careful how you place objects and how you handle fonts and text so that reports and forms look correct on both platforms with nothing cut off or overlapping.

### Placing objects in reports and forms

#### Layout

Keep the following tips in mind when you are placing objects in a report or form:

- ◆ Leave room around the command buttons so that there is room for the thick border the Macintosh uses. On Windows, the border is a slightly darkened line and doesn't take up extra room.
- ◆ Don't put an object in the lower-right corner of a report or form. In Windows, this space can be empty if there is no scrollbar, but on Macintosh this space is always occupied by the size box.

#### Text

Text in your reports and forms is also an important consideration. Since the relative space used by a specific font and size varies on different platforms, you have to take the changes into account when you lay out objects.

FOR INFO For more about text variations, see "Font size" on page 681.

### Using fonts

Both the choice of fonts and the size of fonts can cause problems in reports and forms that will be used on both platforms. Choose fonts that look good in your designs, and verify that the corresponding fonts look good on each platform.

### What font technology to use

You may not be able to control the font technology that others use who share (or run in an application) the objects you create.

**TrueType fonts** TrueType font support is an integrated part of both Macintosh and Windows, so you can use TrueType fonts on both platforms without thinking about fonts. Be sure that each font you use is installed on every computer. Also, be aware that characters are sized differently on Macintosh and Windows.

FOR INFO For more information, see "Font size" on page 681.

**PostScript fonts** If you use PostScript fonts, others will need to have Adobe Type Manager on Windows and Macintosh.

For migration between Windows and Macintosh, InfoMaker provides AccuFonts (from AccuWare Business Solutions Ltd.), which are screen fonts (not printer fonts) for Windows that match the size and appearance of common Macintosh fonts.

FOR INFO For more information, see "AccuFonts for Windows" on page 681.

## When the specified font is not available

When your application uses a font that does not exist on a platform, InfoMaker or the operating system tries to find a substitute.

**Powersoft font mapping file**

InfoMaker provides a font mapping file on each platform that maps commonly-used fonts to fonts that are usually available on that platform. When an application uses a screen or printer font that is unavailable, InfoMaker looks in the [FontSubstitution] section of this file for a mapping. Each line in the file has the following syntax:

*original\_font=mapped\_font*

For example, to map the Times font to the Times New Roman font, the specification is:

Times=Times New Roman

You can change the existing specifications or add your own.

The following sections describe how the substitute font is chosen on each platform.

---

### The font name saved with the object

When you move an object to another platform, the font name stored as the value of an object's property does not change when InfoMaker makes the substitution. The substitution is made on the fly and the saved font name stays the same unless you explicitly change the font.

---

Fonts on Windows

On Windows 95 and Windows NT, the Powersoft font mapping file is called PBFNT60.INI. On Windows 3.1, the file is called PBFNT60W.INI. The font mapping file is stored in the Powersoft SHARED directory. If a font is not available and there is no mapping for it in the font mapping file, Windows itself tries to find a font that matches the characteristics of the missing font. The result may not be satisfactory.

Fonts on the Macintosh

On the Macintosh, the Powersoft font mapping file is called Powersoft Font Preferences and is stored in the System Preferences folder. There is also an internal table that maps standard Windows fonts to Macintosh fonts. The following table shows the font mappings in the internal table:

<b>When the application uses this font and it is not available on Macintosh</b>	<b>InfoMaker substitutes this Macintosh font</b>
System	Chicago
Arial	Helvetica
Courier New	Courier
Helv	Helvetica
MS Sans Serif	Helvetica
MS Serif	Times
Times New Roman	Times
Tms Rmn	Times

InfoMaker reads information from the Powersoft Font Preferences file and passes the result to the internal table.

If the file and the table have different mappings for the same missing font, the mapping in the file is used. For example, if the internal table maps Arial to Helvetica but the file maps Arial to Geneva, Geneva will be substituted for Arial.

If the file maps a missing font to a font that is mapped to another font in the table, the value in the table is substituted for the missing font. For example, if the internal table maps Helv to Helvetica and the file maps Haettenschweiler to Helv, Helvetica will be substituted for Haettenschweiler.

This provides satisfactory results if the Macintosh font is available. If your application uses a less common font and that font is not installed on the Macintosh and not mapped in the preferences file, InfoMaker will use the system font (Chicago).

In addition to the fonts in the internal table, you can use AccuFonts in Windows. AccuFonts provide an exact match for specific Macintosh fonts (see "AccuFonts for Windows" on page 681).

## Font size

Windows and Macintosh each have their own font rendering engine. You'll notice differences in the sizes of fonts in relation to the size of objects when you move a PBL to another platform.

The difference is most noticeable when moving from Windows to Macintosh. In general, if you have used TrueType fonts, you'll find that the same font and point size appears about 30 percent smaller on Macintosh than on Windows.

If you will move a PBL to Windows

If you're creating reports and forms on the Macintosh and plan to move the PBL that contains them to Windows, you should:

- ◆ Leave extra room for text in the objects in reports and forms, both horizontally and vertically.
- ◆ Make columns, labels, and objects, such as buttons in forms, 30 percent smaller to allow for the size increase on Windows.
- ◆ Avoid putting objects too close to the bottom or right side of a report or form.

For example, if you are creating a freeform report or form on the Macintosh, you can move all the columns and labels to the right by one-third the width of the labels and then widen the columns and the labels.

If you will move a PBL to Macintosh

If you are creating a report or form on Windows, you should make your text a little larger than usual so it's not too small on the Macintosh.

## AccuFonts for Windows

AccuFonts for Windows provides bitmap screen fonts that match the size and face of several common Macintosh fonts, such as Geneva and Monaco. AccuFonts is provided by AccuWare Business Solutions Ltd. and comes with InfoMaker for Windows.

Printing with AccuFonts

Because AccuFonts are screen fonts, they don't help with printed output. If you want to display and print the same data, you could consider creating separate reports, one with fonts selected for display, and one with fonts selected for print.

Moving a PBL to Macintosh

If you use AccuFonts in InfoMaker for Windows objects and you move the PBL to the Macintosh, InfoMaker selects the equivalent Macintosh font. The text in your reports and forms will have the same appearance and there will be no problems with size differences.

Moving a PBL to Windows

If you create objects on the Macintosh and move the PBL to Windows, InfoMaker does not use AccuFonts as substitutions for Macintosh fonts.

Selecting a point size

InfoMaker provides the fonts and point sizes listed below. To ensure that your Windows objects will move correctly to the Macintosh, use only the point sizes listed.

---

**Why use the provided point sizes**

If you choose a Accujen at 8 points, which is not provided on the Macintosh, Windows does not scale the font. Although you've selected 8 points, a 9-point font is displayed. The Macintosh will scale the font to 8 points, creating a jagged appearance.

---

<b>Windows font</b>	<b>Equivalent Macintosh font</b>	<b>Available point sizes</b>
Accuat	Athens	18
Accuca	Cairo	18
Accushi	Chicago	12
Accujen	Geneva	9, 10, 12, 14, 18
Accula	Los Angeles	12
Accumo	Mobile	18
Accumon	Monaco	9, 10, 12
Accuny	New York	9, 10, 12, 14, 18
Accusf	San Francisco	18
Accusm *	Symbol	10, 12, 14, 18
Accuve	Venice	12, 14

\* Note that on the Macintosh, Symbol 12 point has a larger line height than Symbol 14 point. Character for character, Accusm matches Apple's Symbol font, but the line heights are different.

## Font settings in the database

When you specify a font in your database as an extended attribute of a column, this font becomes the default for columns in a report or form. Suppose you choose a font that exists on one platform and then move the database to the other platform. If you define more reports or forms on the second platform, you will see the name of the font from the database in the Report or Form painter, but a definition for that font may not exist. To address this issue, you can:

- ◆ Choose another font in the Report or Form painter each time you define a report or form.
- ◆ Change the extended attributes for each column when you move the database to another platform.
- ◆ The extended attributes affect new reports and forms.
- ◆ Select a font that exists on all your target platforms when you initially set up extended attributes.

## Screen resolutions and pixel size

**Windows** Windows systems can have screen resolutions from 640x480 to 1024x768 or greater. For the same screen dimensions, the pixels are larger or smaller depending on the selected resolution. A window will appear larger on lower-resolution screens and smaller on higher-resolution screens.

**Macintosh** Macintosh systems handle pixels in the opposite way. A pixel is always the same size (72 dots per inch) and larger and smaller screens have more and fewer pixels. For example, you have far fewer pixels on a PowerBook screen or a compact Macintosh than you do on a full-size monitor. A window in the lower part of the screen may not even be visible on a PowerBook. (Newer Macintoshes may allow you to select the resolution, making the behavior like Windows.)

## Platform support for InfoMaker features

This table shows InfoMaker features that are supported for Macintosh and Windows.

<b>Feature</b>	<b>Windows</b>	<b>Macintosh</b>
OLE	x	—
Powersoft Reports (PSR files)	x	x
Mail (MAPI)	x	—
AppleScript	—	x
BMP, ICO, CUR images	x	x
WMF images	x	—
PICT images	—	x



# Reference

This part describes the rules for naming objects, the rules for using operators and expressions, and InfoMaker functions.



About this chapter

You use identifiers to name objects. This chapter describes valid identifiers.

Contents

<b>Topic</b>	<b>Page</b>
Rules	688
Reserved words	689

## Rules

### Identifiers:

- ◆ Must start with a letter
- ◆ Can have up to 40 characters, but no spaces
- ◆ Are case insensitive (PART, Part, and part are identical)
- ◆ Can include any combination of letters, numbers, and these special characters:
  - Dash
  - \_ Underscore
  - \$ Dollar sign
  - # Number sign
  - % Percent sign

### Joining words in multiword names

Since InfoMaker does not allow spaces in identifiers, you can use any of the following techniques to join words in an identifier:

- ◆ Initial caps (for example, IncomeJanuary)
- ◆ Dashes (for example, northeast-sales)
- ◆ Underscores (for example, quantity\_on\_hand)

### Examples

Here are some *valid* identifiers:

```
first_quarter_summary
EMPLOYEE_LABELS
EmployeeSalarySummary
Employee_by_#
```

Here are some *invalid* identifiers:

```
2nd-quarter // Does not start with a letter
emp list // Contains a space
Employee'sInfo // Contains an invalid character
```

## Reserved words

You cannot use the following reserved words as identifiers, because InfoMaker uses them internally:

alias	event	not	shared
and	execute	of	static
autoinstantiate	exit	on	step
call	external	open	subroutine
case	false	or	super
choose	fetch	parent	system
close	first	post	systemread
commit	for	prepare	systemwrite
connect	forward	prior	then
constant	from	private	this
continue	function	privateread	to
create	global	privatewrite	trigger
cursor	goto	procedure	true
declare	halt	protected	type
delete	if	protectedread	until
describe	immediate	protectedwrite	update
descriptor	indirect	prototypes	updateblob
destroy	insert	public	using
disconnect	into	readonly	variables
do	intrinsic	ref	while
dynamic	is	return	with
else	last	rollback	within
elseif	library	rpcfunc	_debug
end	loop	select	
enumerated	next	selectblob	



# Operators and Expressions

## About this chapter

You use an expression to request that InfoMaker perform a computational operation. This chapter explains how expressions work and how to write them.

## Contents

<b>Topic</b>	<b>Page</b>
Where you use expressions	692
Operators used in expressions	695
Operator precedence in expressions	701
Matching text patterns	702

## Where you use expressions

An InfoMaker expression is a combination of data, operators, and functions that, when evaluated, results in a value. An expression can include column names, operators, InfoMaker functions, and constants such as numbers and text strings.

**FOR INFO** For information about InfoMaker functions that you can use in expressions, see "Using DataWindow painter and InfoMaker functions" on page 706, or look up the function you want in online Help.

In InfoMaker, you use expressions in these ways:

In this painter	Expressions are used in
Report painter	Computed fields Conditional expressions for object property values Filters Sorting Series and values in graphs Columns, rows, and values in crosstabs
Form painter	Computed fields Conditional expressions for object property values Validation rules
Database painter	Validation rules

---

### Other places where you create expressions

You also use expressions in Quick Select, SQL Select, and the Query painter to specify selection criteria, and in SQL Select and the Query painter to create computed columns. In these painters you are using SQL operators and DBMS-specific functions, not InfoMaker operations and functions, to create expressions. This chapter concerns InfoMaker expressions created with InfoMaker operators and functions.

---

Some of the specific places you use expressions are described here.

#### In computed fields

Expressions for computed fields can evaluate to any value. The data type of the expression becomes the data type of the computed field:

Expression	Description
Today ( )	Displays the date using the Today function
Salary/12	Computes the monthly salary
Sum (Salary for group 1)	Computes the salary for the first group using the Sum aggregate function



Expression	Description
Price*Quantity	Computes the total cost

### Expressions for graphs and crosstabs

You can use similar expressions for series and values in graphs and for columns, rows, and values in crosstabs.

In filters

Filter expressions are boolean expressions that must evaluate to TRUE or FALSE:

Expression	Description
Academics = "*****" AND Cost = "\$\$\$"	Displays data only for colleges with both a 5-star academic rating and a \$\$\$ cost rating
Emp_sal < 50000	Displays data for employees with salaries less than \$50,000
Salary > 50000 AND Dept_id BETWEEN 400 AND 700	Displays data for employees in departments 400, 500, 600, and 700 with salaries greater than \$50,000
Month(Bdate) = 9 OR Month(Bdate) = 2	Displays data for people with birth dates in September or February
Match ( Lname, "[ ^ABC ]" )	Displays data for people whose last name begins with A, B, or C

In validation rules for table columns

Validation rules are boolean expressions that compare column data with values using relational and logical operators. When the validation rule evaluates to FALSE, then the data in the column is rejected.

**In the Form painter** When you specify a validation rule in the Form painter, you want to validate the newly entered value. To refer to the newly entered value, use the `GetText` function. Because `GetText` returns a string, you will also need a data conversion function (such as `Integer` or `Real`) if you compare it to other types of data.

If you include the column name in the expression, you will get the value that already exists for the column instead of the newly entered value that needs validating.

**In the Database painter** When you specify the validation rule in the Database painter, you are defining a general rule that can be applied to any column. Use @placeholder to stand for the newly entered value. The name you use for @placeholder is irrelevant—you can assign the rule to any column that has a data type appropriate for the comparison.

When you define a form, a validation rule assigned to a column is brought into the form and converted to form syntax. @placeholder is converted to GetText and the appropriate data type conversion function.

**Other columns in the rule** You can refer to values in other columns for the current row by specifying their names in the validation rule:

Expression in Database painter	Expression in Form painter	Description
@column >= 10000	Integer(GetText())>= 10000	If a user enters a salary below \$10,000, an error message displays
@column IN (100, 200, 300)	Integer(GetText()) IN (100, 200, 300)	If a user does not enter a department ID of 100, 200, or 300, an error message displays
@salary > 0	Long(GetText()) > 0	If a user doesn't enter a positive number, an error message displays
Match(@disc_price, "[0-9]+\$") and @disc_price < Full_Price	Match(GetText( ), "[0-9]+\$") and Real(GetText()) < Full_Price	If a user enters any characters other than digits, or the resulting number is greater than or equal to the value in the Full_Price column, an error message displays

## Operators used in expressions

An operator is a symbol or word in an expression that performs an arithmetic calculation or logical operation; compares numbers, text, or values; or manipulates text strings.

Reports support the following types of operators:

- ◆ **Arithmetic** for numeric data types
- ◆ **Relational** for all data types
- ◆ **Logical** for all data types
- ◆ **Concatenation** for string data types

### Arithmetic operators

When you write an InfoMaker expression, you can use the following arithmetic operators:

Operator	Meaning	Example
+	Addition	SubTotal + Tax
-	Subtraction	Price - Discount
*	Multiplication	Quantity * Price
/	Division	Discount / Price
^	Exponentiation	Rating ^ 2.5

### Multiplication and division

Multiplication and division are carried out to full precision (16–18 digits). Values are rounded:

Expression	Value
20.0/3	6.666666666666667
3*(20.0/3)	20
Truncate(20.0/3,4)	6.6666

## Calculations with NULL

When you form an arithmetic expression that contains a NULL value, the expression becomes NULL. Thinking of NULL as *undefined* makes this easier to understand.

## Relational operators

You use relational operators to compare a value with other values. The result is a boolean expression whose value is always TRUE or FALSE.

When you write an InfoMaker expression, you can use the following relational operators (for more information about LIKE, IN, and BETWEEN, see after the table):

Operator	Meaning	Example
=	Is equal to	Price = 100
>	Is greater than	Price > 100
<	Is less than	Price < 100
◇	Is not equal to	Price ◇ 100
>=	Greater than or equal to	Price >= 100
<=	Less than or equal to	Price <= 100
NOT =	Is not equal to	Price NOT= 100
LIKE	Matches this specified pattern	Emp_lname LIKE 'C%' OR Emp_lname LIKE 'G%'
IN	Is in this set of values	Dept_id IN (100, 200, 500)
BETWEEN	Is within this range of values. The range includes the first and last values	Price BETWEEN 1000 AND 3000
NOT LIKE	Does not match this specified pattern	Emp_lname NOT LIKE 'C%' AND Emp_lname NOT LIKE 'G%'
NOT IN	Is not in this set of values	Dept_id NOT IN (100, 200, 500)

Operator	Meaning	Example
NOT BETWEEN	Is outside this range of values. The range includes the first and last values	Price NOT BETWEEN 1000 AND 2000

### LIKE and NOT LIKE operators

Use LIKE to search for strings that match a predetermined pattern. Use NOT LIKE to search for strings that do not match a predetermined pattern.

When you use LIKE or NOT LIKE, you can use special characters to match unknown characters in a pattern:

Special character	Meaning	Example
% (percent)	Matches any group of characters	Good% matches all names that begin with Good
_ (underscore)	Matches any single character	Good ___ matches all 7-letter names that begin with Good

For example, the following expression for the Background.Color property of the Salary column displays salaries in red for employees with last names beginning with F and displays all other salaries in white:

```
If(emp_lname LIKE 'F%', RGB(255,0,0), RGB(255,255,255))
```

### BETWEEN and NOT BETWEEN operators

Use BETWEEN to check if a value is within a range of values. Use NOT BETWEEN to check if a value is *not* in a range of values. The range of values includes the boundary values that specify the range.

For example, the following expression for the Background.Color property of the Salary column displays salaries in red when an employee's salary is between \$50,000 and \$100,000 and displays all other salaries in white:

```
If(salary BETWEEN 50000 AND 100000, RGB(255,0,0),
   RGB(255,255,255))
```

### IN and NOT IN operators

Use IN to check if a value is in a set of values. Use NOT IN to check if a value is *not* in a set of values.

For example, the following expression for the Background.Color property of the Salary column displays salaries in red for employees in department 300 or 400 having a salary between \$50,000 and \$100,000, and displays all other salaries in white:

```
If(dept_id IN (300,400) and salary BETWEEN 50000 AND
   100000, RGB(255,0,0), RGB(255,255,255))
```

## Logical operators

You use logical operators to combine boolean expressions into a larger boolean expression. The result is always TRUE or FALSE:

Operator	Meaning	Example
NOT	Logical negation If A is true, NOT A is false If A is false, NOT A is true	NOT Price = 100
AND	Logical and A AND B is true if both are true A AND B is false if either is false	Tax > 3 AND Ship < 5
OR	Logical or A OR B is true if either is true or both are true A OR B is false only if both are false	Tax > 3 OR Ship < 5

When you combine two or more boolean expressions to form a new expression, the new expression is either true or false. The following truth table shows how TRUE and FALSE expressions are evaluated to form an expression that is either TRUE or FALSE.

For example, if "My dog has fleas" is true and "My hair is brown" is false, then "My dog has fleas OR my hair is brown" is true, and "My dog has fleas AND my hair is brown" is false:

If one expression has this value	And the logical operator is	And if another expression has this value	The resulting expression has this value
TRUE	AND	TRUE	TRUE
TRUE	AND	FALSE	FALSE
FALSE	AND	TRUE	FALSE
FALSE	AND	FALSE	FALSE
TRUE	OR	TRUE	TRUE
TRUE	OR	FALSE	TRUE
FALSE	OR	TRUE	TRUE
FALSE	OR	FALSE	FALSE
NOT TRUE	AND	TRUE	FALSE

If one expression has this value	And the logical operator is	And if another expression has this value	The resulting expression has this value
NOT TRUE	AND	FALSE	FALSE
NOT FALSE	AND	TRUE	TRUE
NOT FALSE	AND	FALSE	FALSE
NOT TRUE	OR	TRUE	TRUE
NOT TRUE	OR	FALSE	FALSE
NOT FALSE	OR	TRUE	TRUE
NOT FALSE	OR	FALSE	TRUE

## Comparing strings

When you compare strings, the comparison is case sensitive. Trailing blanks are significant.

### Case-sensitivity examples

Assume City1 is "Austin" and City2 is "AUSTIN". Then:

```
City1=City2
```

returns FALSE.

To compare strings regardless of case, use the Upper or Lower function. For example:

```
Upper(City1)=Upper(City2)
```

returns TRUE.

**FOR INFO** For information about these functions, see "Using DataWindow painter and InfoMaker functions" on page 706.

### Trailing blanks examples

Assume City1 is "Austin" and City2 is "Austin ". Then the expression:

```
City1=City2
```

returns FALSE.

To prevent blanks from affecting a comparison, remove them with one of the Trim functions: Trim, LeftTrim, or RightTrim.

For example:

```
Trim(City1)=Trim(City2)
```

returns TRUE.

FOR INFO For information about these functions, see "Using DataWindow painter and InfoMaker functions" on page 706.

## Concatenation operator

The concatenation operator joins the contents of two variables of the same type to form a longer value. You can concatenate strings and blobs.

To concatenate values, you use the plus sign (+) operator.

String expression	Value
"over" + "stock"	overstock
Lname + ', ' + Fname	If Lname is Hill and Fname is Craig, then "Hill, Craig"

---

### Using quotes

You can use either single or double quotes in string expressions. For example, the expression "over" + "stock" is equivalent to the expression 'over'+'stock'.

---



## Operator precedence in expressions

To ensure predictable results, all operators in an expression are evaluated in a specific order of precedence. When the operators have the same precedence, InfoMaker evaluates them left to right.

The following table lists the operators in descending order of precedence:

Operator	Purpose
()	Grouping
^	Exponentiation
*, /	Multiplication and division
+, -	Addition and subtraction; string concatenation
IN, LIKE, BETWEEN	SQL SELECT statement conditions
NOT	Logical negation
=, >, <, <=, >=, <>	Relational operators
AND, OR	Logical and and logical or

## Overriding the precedence order

Since expressions in parentheses are evaluated first, to override the precedence order, enclose expressions in parentheses. Within each set of parentheses, precedence order applies.

In the expression  $x+y*a+b$ ,  $y$  is first multiplied by  $a$  (because multiplication has a higher precedence than addition). The result of the multiplication is then added to  $x$  and this result is then added to  $b$  (because the  $+$  operators are evaluated left to right).

To force evaluation in a different order, group expressions with parentheses. For example, in the expression  $x+(y*(a+b))$ ,  $a+b$  is evaluated first. The sum  $a+b$  is then multiplied by  $y$ , and this product is added to  $x$ .

## Matching text patterns

A text pattern is an expression that you can use to evaluate whether a string contains a particular pattern of characters. Text patterns consist of metacharacters, which have special meaning, and characters (nonmetacharacters). The way the metacharacters and normal characters are combined specify the pattern you are looking for.

### Metacharacters

Valid metacharacters are:

- ◆ ^ (caret)
- ◆ \$ (dollar sign)
- ◆ . (period)
- ◆ \ (backslash)
- ◆ [ ] (brackets)
- ◆ \* (asterisk)
- ◆ + (plus)
- ◆ ? (question mark)

### Nonmetacharacters

In a text pattern, you can also use characters that are not metacharacters. One or more nonmetacharacters in a text pattern (such as letters) match the characters themselves. For example, A matches A, and ABC matches ABC.

### Examples

Text patterns are needed in all expressions that use the Match function. Text patterns are used in validation rules in the Form painter (in a column object's Validation property page) and the Database painter (in the Match Pattern dialog box).

**FOR INFO** For information about the Match function, see Match on page 786.

The following table shows various text patterns and sample text that matches each pattern:

This pattern	Matches
AB	Any string that contains AB; for example, ABA, DEABC, graphAB_one
B*	Any string that contains 0 or more Bs; for example, AC, B, BB, BBB, ABBBC, and so on
AB*C	Any string containing the pattern AC or ABC or ABBC, and so on (0 or more Bs)

<b>This pattern</b>	<b>Matches</b>
AB+C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (1 or more Bs)
ABB*C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (1 B plus 0 or more Bs)
^AB	Any string starting with AB
AB?C	Any string containing the pattern AC or ABC (0 or 1 B)
^[ABC]	Any string starting with A, B, or C
[^ABC]	A string containing any characters other than A, B, or C
^[^abc]	A string that begins with any character except a, b, or c
^[^a-z]\$	Any single-character string that is not a lowercase letter (^ and \$ indicate the beginning and end of the string)
[A-Z]+	Any string with one or more uppercase letters
^[0-9]+\$	Any string consisting only of digits
^[0-9][0-9][0-9]\$	Any string consisting of exactly three digits
^([0-9][0-9][0-9])\$	Any string consisting of exactly three digits enclosed in parentheses



# DataWindow Painter and InfoMaker Functions

About this chapter

This chapter provides syntax, descriptions, and examples of the functions you can use in expressions in PowerBuilder's DataWindow painter and in InfoMaker's Report painter and Form painter.

Contents

After a short introduction and several examples, the functions are listed alphabetically.

## Using DataWindow painter and InfoMaker functions

PowerBuilder DataWindow functions and InfoMaker functions are the same functions. In PowerBuilder's DataWindow painter and InfoMaker's Report and Form painters, you can use these functions in expressions for computed fields, filters, validation rules, and graphed data, with some exceptions.

The dialog boxes in which you define these expressions include a listbox that lists the available functions and their arguments. The dialog boxes make it easy to insert the function into the expression.

**FOR INFO** For information about expressions, see Chapter 22, "Operators and Expressions".

Return values for functions and expressions

DataWindow painter and InfoMaker expressions can return the following data types:

- ◆ Double
- ◆ String
- ◆ DateTime
- ◆ Time

Within an expression, a function can return other data types (such as boolean, date, or integer) but the final value of an expression is converted to one of the four data types.

Restrictions for aggregate functions

An aggregate function is a function (such as Avg, Max, StDev, and Sum) that operates on a range of values in a column. When you use an aggregate function, some restrictions apply. You cannot use an aggregate function:

- ◆ In a filter
- ◆ In a validation rule
- ◆ As an argument for another aggregate function

When you use aggregate functions, they cancel the effect of setting Retrieve Rows As Needed. To do the aggregation, the report always retrieves all rows.

Formatting for the locally correct display of numbers

No matter what country you are creating objects and developing an application in, you must use U.S. number notation in numbers or number masks in display formats, edit masks, and DataWindow painter and InfoMaker expressions. This means that when you specify a number or number mask, use a comma as the thousands delimiter and period for the decimal place.

Numbers will display appropriately in whatever countries you deploy applications in. During execution, the locally correct symbols for numbers will display (because PowerBuilder and InfoMaker use the international Control Panel settings) when numbers are interpreted. For example, in countries where comma represents the decimal place and period represents thousands, users will see numbers in those formats during execution.

**FOR INFO** For information about the locally correct display of dates and day names, see `String` on page 845 and `DayName` on page 751.

## Four examples

### Example 1: counting NULL values in a column

A NULL value is a marker used to fill a place in a column where data is missing for any reason. The value may not be applicable, or it may be missing or unknown. When a database table is created, each column in the table either allows NULL values or does not allow them. The column or set of columns that define the primary key cannot allow NULL values. Sometimes it's useful to know how many NULL values there are in a particular column.

What you want to do

You are working with the Fin\_code table in the Powersoft Demo Database. The Fin\_code table has three columns:

Column	What the column is	Allows NULL values?
Code	Unique financial identifier (primary key)	No
Type	Code type: expense or revenue	No
Description	Code description: the department incurring the expense or getting the revenue	Yes

You create a report using the Code and Description columns. You want to know the number of NULL values in the Description column.

How to do it

In the report, you create a computed field that uses functions to display the number of NULL values in the Description column.

For the sake of demonstrating the use of functions, the following computed fields are created in the Summary band of the report (with text objects that tell you what information each computed field is providing):

```
Count(description for all)
```

which counts the number of descriptions (that are not NULL);

```
Sum(If(IsNull(description), 1, 0))
```

which returns a 1 if the description column is NULL, a 0 if the description column is NOT NULL, and then adds the total;

```
Count(id for all)
```

which counts the number of IDs (which is also the number of rows);



```
Sum(If(IsNull(description), 1, 1))
```

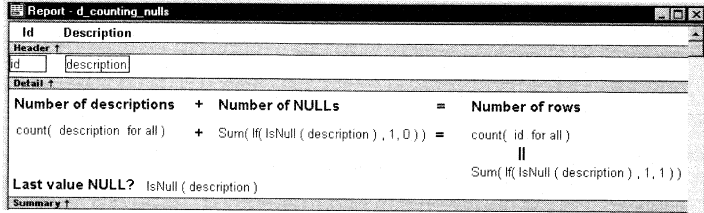
which adds the number of NULLs and NOT NULLs in the description column (which is the total number of rows) and should match the result of the Count( id for all ) function; and

```
IsNull(description)
```

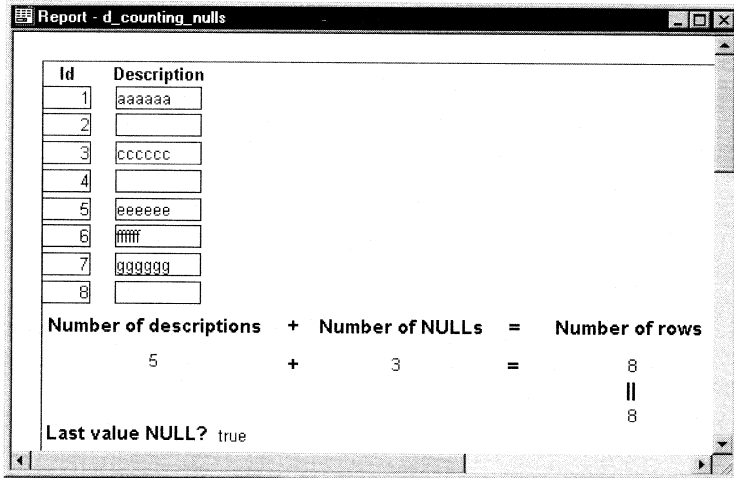
which evaluates whether the last row in the table has a description that is NULL. The return value of the IsNull function is TRUE or FALSE.

What you get

Here's the design for the report.



Here's the report. The report or report shows there are eight descriptions, three of which are NULL and five of which are not NULL. The last description for Id=8 is NULL.



## Example 2: counting male and female employees

Example 1 demonstrates the use of the Sum and Count functions. Sum and Count are two examples of a class of functions called aggregate functions.

An aggregate function is a function that operates on a range of values in a column. The aggregate functions are:

Avg	Large	Mode	Sum
Count	Last	Percent	Var
CumulativePercent	Max	Small	VarP
CumulativeSum	Median	StDev	
First	Min	StDevP	

---

### About crosstab functions

Although the crosstab functions (CrosstabAvg, CrosstabCount, CrosstabMax, CrosstabMin, and CrosstabSum) behave like aggregate functions, they are not included on the list because they are for crosstabs only and are designed to work in the crosstab matrix.

---

A few restrictions apply to the use of aggregate functions. You can't use an aggregate function:

- ◆ In a filter
- ◆ In a validation rule
- ◆ As an argument for another aggregate function

This example demonstrates the use of the Sum aggregate function.

What you want to do

Using the Employee table in the Powersoft Demo Database as the data source, you create a report using at least the Emp\_id and the Sex columns. You want the report to display the number of male employees and female employees in the company.

How to do it

In the summary band in the workspace, add two computed fields to the report that use the Sum and If functions:

```
Sum(If(sex = "M", 1, 0))
```

which counts the number of males in your company;

```
Sum(If(sex = "F", 1, 0))
```

which counts the number of females in your company.

You can also add a Page computed field (by clicking the Page computed field button) in the footer band to display the page number and total pages at the bottom of each page of the report.

What you get

Here's what the design of the report looks like.

Report - counting_gender	
Employee ID	Sex
<b>Header ↑</b>	
emp_id	<input type="radio"/> Male <input type="radio"/> Female
<b>Detail ↑</b>	
<b>Number of males</b>	<b>Number of females</b>
Sum ( If (sex = "M", 1, 0) )	Sum ( If (sex = "F", 1, 0) )
<b>Summary ↑</b>	
'Page ' + page() + ' of ' + pageCount()	
<b>Footer ↑</b>	

Here's the last page of the report in preview, with the total number of males and females in the company displayed.

1684	<input type="radio"/> Male	
	<input checked="" type="radio"/> Female	
1740	<input checked="" type="radio"/> Male	
	<input type="radio"/> Female	
1751	<input checked="" type="radio"/> Male	
	<input type="radio"/> Female	
<b>Number of males</b>		<b>Number of females</b>
41		34
Page 3 of 3		

If you now want more information

What if you decide that you also want to know the number of males and females in each department in the company?

❖ **To display the males and females in each department:**

- 1 Click the SQL button so you can edit the data source.
- 2 Open the Department table in the SQL painter workspace, which currently displays the Employee table with the Emp\_id and Sex columns selected.
- 3 Select the department\_dept\_name column to add it to your data source.
- 4 Select Rows>Create Group from the menu bar to create a group and group by department name.

- 5 In the trailer group band in the workspace, add two additional computed fields to your report:

`Sum(If(sex = "M", 1, 0) for group 1)`

which counts the number of males in each department;

`Sum(If(sex = "F", 1, 0) for group 1)`

which counts the number of females in each department.

Here's what the design of the grouped report looks like.

Report - counting_gender_by_department	
<b>Employee ID</b>	<b>Sex</b>
Header ↑	
department_dept_name	
1: Header group department_dept_name ↑	
emp_id	<input type="radio"/> Male <input type="radio"/> Female
Detail ↑	
<b>Number of males</b>	<b>Number of females</b>
Sum ( If (sex = "M", 1, 0) for group 1 )	Sum ( If (sex = "F", 1, 0) for group 1 )
1: Trailer group department_dept_name ↑	
<b>Total number of males</b>	<b>Total number of females</b>
Sum ( If (sex = "M", 1, 0) )	Sum ( If (sex = "F", 1, 0) )
Summary ↑	
'Page ' + page() + ' of ' + pageCount()	
Footer ↑	

Here's the last page of the report with the number of males and females in the shipping department displayed, followed by the total number of males and females in the company.

Shipping	
191	<input type="radio"/> Male <input checked="" type="radio"/> Female
703	<input checked="" type="radio"/> Male <input type="radio"/> Female
750	<input type="radio"/> Male <input checked="" type="radio"/> Female
868	<input type="radio"/> Male <input checked="" type="radio"/> Female
321	<input checked="" type="radio"/> Male <input type="radio"/> Female
1013	<input checked="" type="radio"/> Male <input type="radio"/> Female
1570	<input checked="" type="radio"/> Male <input type="radio"/> Female
1615	<input type="radio"/> Male <input checked="" type="radio"/> Female
1658	<input checked="" type="radio"/> Male <input type="radio"/> Female
<b>Number of males</b>	<b>Number of females</b>
5	4
<b>Total number of males</b>	<b>Total number of females</b>
41	34

### Example 3: creating a row indicator

This example demonstrates the use of several functions: `Bitmap`, `Case`, `CurrentRow`, `GetRow`, and `RGB`.

The example is presented in PowerBuilder's DataWindow painter, which is the same as InfoMaker's Report painter. You can use all the functions shown in the example in the Report painter. However, because you can change the current row and change data in a DataWindow object (which you can't do in a report), the example is more interesting to consider in a DataWindow object.

What you want to do

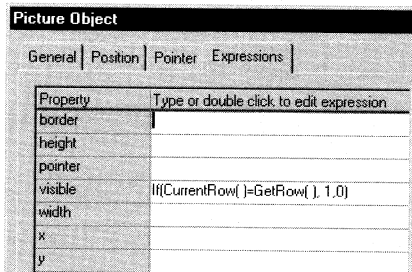
Using the Employee table in the Powersoft Demo Database, you create a DataWindow object using the `Emp_id`, `Emp_fname`, `Emp_lname`, and `Salary` columns.

In the DataWindow painter, you want to display a number of items such as the number of the current row, an arrow that is an indicator of the current row, and the salary for an employee with a background color that depends on what the salary is.

How to do it

In the workspace, add the following:

- ◆ A computed field `CurrentRow()` which displays the number of the current row
- ◆ A picture object, which is a right-arrow, for which you define (in the Expressions property page of the Picture Object property sheet) an expression for the arrow's visible property:



The expression results in an arrow displaying in the current row and no arrow displaying in other rows.

- ◆ A computed field using the `If`, `CurrentRow`, and `GetRow` functions:

```
If(CurrentRow() = GetRow(), "Current", "Not current")
```

which displays the word "Current" when the row is the current row and "Not current" for all other rows

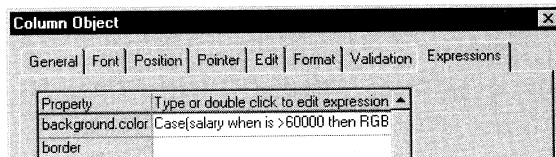
- ◆ A computed field (typed on one line) using the Bitmap, CurrentRow, and GetRow functions:

```
Bitmap(If(CurrentRow()=GetRow(),
" c:\pb5i32\ex\code\indicatr.bmp", " "))
```

which displays an arrow bitmap for the current row and no bitmap for all other rows

- ◆ An expression (on one line in the Expressions property page of the Column Object property sheet) for the Background.Color property of the salary column:

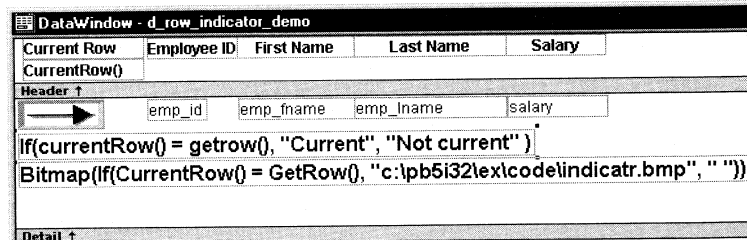
```
Case(salary WHEN IS >60000 THEN RGB(192,192,192)
WHEN IS >40000 THEN RGB(0,255,0) ELSE
RGB(255,255,255))
```



The expression results in a salary above \$40,000 displaying in green, a salary above \$60,000 displaying in gray, and all other salaries displaying in white.

What you get

Here's what the design of the DataWindow object looks like:



After previewing the DataWindow object, here's what the data looks like with the second row current.

Current Row	Employee ID	First Name	Last Name	Salary
2	102	Fran	Whitney	\$45,700.00
<b>Not current</b>				
	105	Matthew	Cobb	\$62,000.00
<b>Current</b>				
	129	Philip	Chin	\$38,500.00
<b>Not current</b>				

Notice that the number of the current row is 2; the first row and the third row are "Not current" (and therefore display no bitmap); and the second row, which is the current row, displays the arrow row indicator.

On your screen, the salary in the first row has a green background because it's more than \$40,000; the salary in the second row has a gray background because it's more than \$60,000; and the salary in the third row has a white background, which matches the background of the DataWindow object.

## Example 4: displaying all data when a column allows NULLS

When you create an arithmetic expression that has a NULL value, the value of the expression is NULL. This makes sense, since NULL means essentially undefined and the expression is undefined. But sometimes this fact can interfere with what you want to display.

What you want to do

A table in your database has four columns: Id, Corporation, Address1, and Address2. The Corporation, Address1, and Address2 columns allow NULLS. Using this table as the data source, you create a report using the four columns. You now want the report to display both parts of the address, separated by a comma.

You create a computed field to concatenate Address1 and Address2 with a comma separator. Here's the expression that defines the computed field:

```
address1 + ", " + address2
```

When you preview the report, if either Address1 or Address2 is NULL, no part of the address displays—because the value of the expression is NULL. To display a part of the address, you need to create a computed field that forces evaluation even if Address2 is NULL. Note that we assume that Address2 will only have data if Address1 has data for a particular row.

How to do it

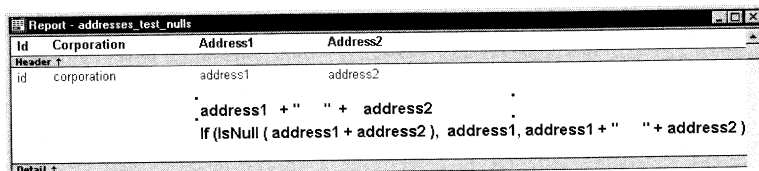
In the detail band, create a computed field that uses the If and IsNull functions:

```
If(IsNull(address1 + address2), address1, address1 +
", " + address2)
```

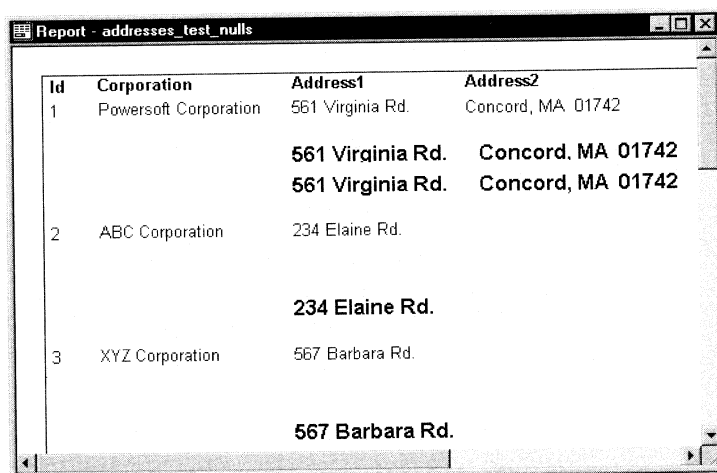
The computed field says this: if the concatenation of the addresses is NULL (because address2 is NULL), then display address1, and if it's not NULL, display both parts of the address separated by a comma.

What you get

Here's what the design of the report looks like. It includes both the computed field that doesn't work and the one that does.



When you preview the report, notice that the first computed field displays NULL for ABC Corporation and XYZ Corporation. The second computed field displays the first part of the address, which is not NULL.





## Other examples

In InfoMaker, to see some examples of using functions, examine the reports and forms in TUTOR\_IM.PBL, which is InfoMaker's sample library. The reports and forms were created using data in the Powersoft Demo Database.

Look carefully at the reports whose names begin with *attrib\_*. Each report is a good example of the use of functions in the Expressions property page in an object's property sheet. And look at the design of each report and form in the sample library to see the use of functions in other ways.

For more information

For examples of using the Expressions property page in an object's property sheet, see Chapter 8, "Highlighting Information in Reports and Forms".

For an example of the use of each InfoMaker function, see the function descriptions that follow.

## Abs

Description                      Calculates the absolute value of a number.

Syntax                              **Abs** ( *n* )

Argument	Description
<i>n</i>	The number for which you want the absolute value

Return value                      The data type of *n*. Returns the absolute value of *n*.

Examples                            This expression counts all the product numbers where the absolute value of the product number is distinct:

```
Count(product_number for All DISTINCT Abs
      (product_number))
```

Only data with an absolute value greater than 5 passes this validation rule:

```
Abs(value_set) > 5
```

See also                            **Count**

## Asc

**Description** Converts the first character of a string to its ASCII integer value.

**Syntax** **Asc** ( *string* )

Argument	Description
<i>string</i>	The string for which you want the ASCII value of the first character

**Return value** Integer. Returns the ASCII value of the first character in *string*.

**Usage** Use Asc to test the case of a character or manipulate text and letters.

To find out the case of a character, you can check whether its ASCII value is within the appropriate range.

**Examples** This expression for a computed field returns the string in `code_id` if the ASCII value of the first character in `code_id` is A (65):

```
If (Asc(code_id) = 65, code_id, "Not a valid code")
```

This expression for a computed field checks the case of the first character of `lname` and if it is lowercase makes it uppercase:

```
IF (Asc(lname) > 64 AND Asc(lname) < 91, lname,
WordCap(lname))
```

**See also** Char  
WordCap

# Avg

Description

Calculates the average of the values of the column.

Syntax

**Avg** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the average of the data values. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the average. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The average of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The average of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The average of the values in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The average of all values in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The average of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The average of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Avg to consider only the distinct values in <i>column</i> when calculating the average. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Return value

The numeric data type of the column. Returns the average of the values of the rows in *range*.

## Usage

If you specify *range*, Avg returns the average value of *column* in *range*. If you specify DISTINCT, Avg returns the average value of the distinct values in *column*, or if you specify *expressn*, the average of *column* for each distinct value of *expressn*.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

In calculating the average, NULL values are ignored.

---

#### Not in validation rules or filter expressions

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

## Examples

This expression returns the average of the values in the column named salary:

```
Avg(salary)
```

This expression returns the average of the values in group 1 in the column named salary:

```
Avg(salary for group 1)
```

This expression returns the average of the values in column 5 on the current page:

```
Avg(#5 for page)
```

This computed field returns Above Average if the average salary for the page is greater than the average salary:

```
If(Avg(salary for page) > Avg(salary), "Above  
Average", " ")
```

This expression for a graph value sets the data to the average value of the sale\_price column:

**Avg**(sale\_price)

This expression for a graph value sets the data value to the average value of the sale\_price column for the entire graph:

**Avg**(sale\_price for graph)

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the average of the order amount for the distinct order numbers:

**Avg**(order\_amt for all DISTINCT order\_nbr)

See also

Median  
Mode

# Bitmap

**Description** Displays the specified bitmap.

---

**For computed fields only**

You can use the Bitmap function *only* in a computed field.

---

**Syntax**

**Bitmap** ( *string* )

Argument	Description
<i>string</i>	A column containing bitmap files, a string containing the name of an image file (a BMP, RLE, or WMF file), or an expression that evaluates to a string containing the name of an image file

**Return value**

The special data type bitmap, which *cannot* be used in any other function.

**Usage**

Use Bitmap to dynamically display a bitmap in a computed field. When *string* is a column containing bitmap files, a different bitmap can display for each row.

**Examples**

These examples are all expressions for a computed field.

This expression dynamically displays the bitmap file contained in the column named employees:

```
Bitmap (employees)
```

If the employees column is column 3, this next expression gives the same result as the expression above:

```
Bitmap (#3)
```

This expression displays the bitmap TOOLS.BMP:

```
Bitmap ("TOOLS.BMP")
```

This expression tests the value in the column named password and then uses the value to determine which bitmap to display:

```
Bitmap (If (password = "y", "yes.bmp", "no.bmp"))
```

## Case

**Description** Tests the values of a column or expression and returns values based on the results of the test.

**Syntax** **Case** ( *column* WHEN *value1* THEN *result1* { WHEN *value2* THEN *result2* { ... } } { ELSE *resultelse* } )

Argument	Description
<i>column</i>	The column or expression whose values you want to test. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. <i>Column</i> is compared to each <i>valuen</i>
WHEN (optional)	Introduces a value-result pair. At least one WHEN is required
<i>valuen</i>	One or more values that you want to compare to values of <i>column</i> . A value can be: <ul style="list-style-type: none"> <li>◆ A single value</li> <li>◆ A list of values separated by commas (for example, 2, 4, 6, 8)</li> <li>◆ A TO clause (for example, 1 TO 20)</li> <li>◆ IS followed by a relational operator and comparison value (for example, IS&gt;5)</li> <li>◆ Any combination of the above with an implied OR between expressions (for example, 1,3,5,7,9,27 TO 33, IS&gt;42)</li> </ul>
THEN	Introduces the result to be returned when <i>column</i> matches the corresponding <i>valuen</i>
<i>resultn</i>	An expression whose value is returned by Case for the corresponding <i>valuen</i> . All <i>resultn</i> values must have the same data type
ELSE (optional)	Specifies that for any values of <i>column</i> that don't match the values of <i>valuen</i> already specified, Case returns <i>resultelse</i>
<i>resultelse</i>	An expression whose value is returned by Case when the value of <i>column</i> doesn't match any WHEN <i>valuen</i> expression

**Return value** The data type of *resultn*. Returns the result you specify in *resultn*.

**Usage** If more than one WHEN clause matches *column*, Case returns the result of the first matching one.



**Examples**

This expression for the Background.Color property of a Salary column returns values that represent red when an employee's salary is greater than \$70,000, green when an employee's salary is greater than \$50,000, and blue otherwise:

```
Case(salary WHEN IS >70000 THEN RGB(255,0,0) WHEN IS  
>50000 THEN RGB(0,255,0) ELSE RGB(0,0,255))
```

This expression for the Background.Color property of an employee Id column returns red for Id 101, gray for Id 102, and black for all other Id numbers:

```
Case(emp_id WHEN 101 THEN 255 WHEN 102 THEN  
RGB(100,100,100) ELSE 0)
```

This expression for the Format property of the Marital\_status column returns Single, Married, and Unknown based on the data value of the Marital\_status column for an employee:

```
Case(marital_status WHEN 'S' THEN 'Single' WHEN 'M'  
THEN 'Married' ELSE 'Unknown')
```

See also

If

# Ceiling

**Description** Retrieves the smallest whole number that is greater than or equal to a specified limit.

**Syntax** **Ceiling** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The number for which you want the smallest whole number that is greater than or equal to it

**Return value** The data type of *n*. Returns the smallest whole number that is greater than or equal to *n*.

**Examples** These expressions both return - 4:

```
Ceiling(-4.2)
```

```
Ceiling(-4.8)
```

This expression for a computed field returns **ERROR** if the value in `discount_amt` is greater than the smallest whole number that is greater than or equal to `discount_factor` times price. Otherwise, it returns `discount_amt`:

```
If(discount_amt <= Ceiling(discount_factor * price),  
String(discount_amt), "ERROR")
```

To pass this validation rule, the value in `discount_amt` must be less than or equal to the smallest whole number that is greater than or equal to `discount_factor` times price:

```
discount_amt <= Ceiling(discount_factor * price)
```

**See also**

**Int**  
**Round**  
**Truncate**

# Char

Description Converts an integer to a character.

Syntax **Char** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The integer you want to convert to a character

Return value String. Returns the character whose ASCII value is *n*.

Examples This expression returns the escape character:

**Char** (27)

See also Asc

## Cos

Description                      Calculates the cosine of an angle.

Syntax                              **Cos** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The angle (in radians) for which you want the cosine

Return value                      Double. Returns the cosine of *n*.

Examples                            This expression returns 1:

**Cos** ( 0 )

This expression returns .540302:

**Cos** ( 1 )

This expression returns - 1:

**Cos** ( Pi ( 1 ) )

See also                            Pi  
Sin  
Tan

# Count

Description

Calculates the total number of rows in the specified column.

Syntax

**Count** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } }

Argument	Description
<i>column</i>	The column for which you want the number of rows. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column
FOR <i>range</i> (optional)	The data that will be included in the count. For most presentation styles, values for <i>range</i> are: <ul style="list-style-type: none"> <li>◆ ALL — (Default) The count of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The count of rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The count of the rows in <i>column</i> on a page</li> </ul> For Crosstabs, specify CROSSTAB for <i>range</i> : <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The count of all rows in <i>column</i> in the crosstab</li> </ul> For Graph and OLE objects, specify the type of object for <i>range</i> . The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are: <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The count of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The count of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Count to consider only the distinct values in <i>column</i> when counting the rows. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Usage

If you specify *range*, Count determines the number of rows in *column* in *range*. If you specify DISTINCT, Count returns the number of the distinct rows displayed in *column*, or if you specify *expresn*, the number of rows displayed in *column* where the value of *expresn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

Null values in the column are ignored and are not included in the count.

---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or a report always retrieves all rows.

**Examples**

This expression returns the number of rows in the column named emp\_id that are not NULL:

```
Count(emp_id)
```

This expression returns the number of rows in the column named emp\_id of group 1 that are not NULL:

```
Count(emp_id for group 1)
```

This expression returns the number of dept\_ids that are distinct:

```
Count(dept_id for all DISTINCT)
```

This expression returns the number of regions with distinct products:

```
Count(region_id for all DISTINCT Lower(product_id))
```

This expression returns the number of rows in column 3 on the page that are not NULL:

```
Count(#3 for page)
```

## CrosstabAvg

**Description** Calculates the average of the values returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabAvg can also calculate averages of the expression's values for groups of column values.

---

### For crosstabs only

You can use this function *only* in a crosstab DataWindow object or report.

---

### Syntax

**CrosstabAvg** ( *n* { *column*, *groupvalue* } )

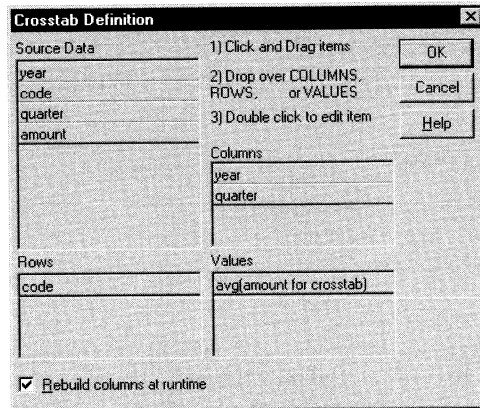
Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the average of the returned values. The crosstab expression must be numeric
<i>column</i> (optional)	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations
<i>groupvalue</i> (optional)	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name as a quoted string

**Return value** Double. Returns the average of the crosstab values returned by expression *n* for all the column values or, optionally, for a subset of column values.

**Usage** This function is meaningful *only* for the average of the values of the expression in a *row* in the crosstab. This means you can use it only in the detail band, not in a header, trailer, or summary band.

NULL values are ignored and are not included in the average.

**How PowerBuilder and InfoMaker use the functions in a crosstab** When PowerBuilder and InfoMaker generate a crosstab from your definition, they automatically create the appropriate computed fields using the Crosstab functions. To understand the functions, consider a crosstab with two columns (year and quarter) a row (product), and the values expression Avg(amount for crosstab). The Crosstab Definition dialog box looks like this.



When you define the crosstab described above, the DataWindow painter or Report painter automatically creates the appropriate computed fields. A computed field named avg\_amount returns the average of the quarterly figures for each year. Its expression is:

```
CrosstabAvg(1, 2, "@year")
```

A second computed field named grand\_avg\_amount computes the average of all the amounts in the row. Its expression is:

```
CrosstabAvg(1)
```

Other computed fields in the summary band use the Avg function to display the average of the values in the amount column, the yearly averages, and the final average.



The crosstab in the DataWindow painter or Report painter looks like this.

	Year	Quarter	
<b>Header [1] ↑</b>	@year	@year Avg	
<b>Header [2] ↑</b>			
<b>Code</b>	@quarter		Grand Avg
<b>Header [3] ↑</b>			
code	amount	crosstabavg(1, 2, "@year")	crosstabavg(1)
<b>Detail ↑</b>			
"Grand Avg"	avg(amount for all )	avg(avg_amount for all )	avg(grand_avg_amount for all )
<b>Summary ↑</b>			
<b>Footer ↑</b>			

Each row in the crosstab (after adjusting the column widths) has cells for the amounts in the quarters, a repeating cell for the yearly average, and a grand average. The crosstab also displays averages of the amounts for all the financial codes in the quarters in the summary band at the bottom.

	Year				Quarter					
	1992				1992 Avg	1993				1993 Avg
Code	Q1	Q2	Q3	Q4		Q1	Q2	Q3	Q4	
e1	101	93	129	145	117	153	149	157	163	156
e2	403	459	609	632	526	643	687	896	923	788
e3	1,437	2,033	2,184	2,145	1,950	2,478	2,998	3,702	3,600	3,195
e4	623	784	856	1,043	827	1,051	1,158	1,459	1,439	1,277
e5	381	402	412	467	416	523	749	723	748	686
r1	1,023	2,033	2,998	3,014	2,267	3,114	3,998	6,523	7,267	5,226
r2	234	459	601	944	560	992	1,195	1,704	1,823	1,429
<b>Grand Avg</b>	<b>600</b>	<b>895</b>	<b>1,113</b>	<b>1,199</b>	<b>952</b>	<b>1,279</b>	<b>1,562</b>	<b>2,167</b>	<b>2,280</b>	<b>1,822</b>

1994					1994 Avg	
Q1	Q2	Q3	Q4		Grand Avg	
198	204	214	231		212	161
921	975	984	982		966	760
4,139	4,500	4,532	5,298		4,617	3,254
1,462	1,472	1,439	1,498		1,468	1,190
798	983	956	963		925	675
9,144	10,988	13,567	15,199		12,225	6,572
1,839	2,011	2,897	4,129		2,719	1,569
<b>2,643</b>	<b>3,019</b>	<b>3,513</b>	<b>4,043</b>		<b>3,304</b>	<b>2,026</b>

**What the function arguments mean** When the crosstab definition has more than one column, you can specify column qualifiers for any of the Crosstab functions, so that the crosstab displays calculations for groups of column values. As illustrated previously, when year and quarter are the columns in the crosstab the expression for the computed field is:

`CrosstabAvg(1, 2, "@year")`

The value 2 refers to the quarter column (the second column in the Crosstab Definition dialog) and "@year" specifies grouping values from the year column (meaning the function will average values for the quarters within each year). The value 1 refers to the crosstab-values expression that will be averaged. In the resulting crosstab, the computed field repeats in each row after the cells for the quarters within each year.

**Tips for defining crosstabs** When you define a crosstab with more than one column, the order of the columns in the Columns box of the Crosstab Definition dialog box governs the way PowerBuilder and InfoMaker group the columns. To let PowerBuilder and InfoMaker define the most effective expressions, make the column that contains the grouping values (for example, year or department) the first column in the Columns box and the column that contains the values to be grouped (for example, quarter or employee) second.

To display calculations for groups of rows, define groups as you would for other DataWindow objects or report presentation styles and define computed fields in the group header or footer using noncrosstab aggregation functions, such as Avg, Sum, or Max.

---

### Reviewing the expressions

To review the expressions defined for the crosstab values, open the Crosstab Definition dialog box (display the popup menu in an unused area of the crosstab and then select Crosstab).

---

### Examples

The first two examples use the crosstab expressions shown below:

```
Count(emp_id for crosstab),Sum(salary for crosstab)
```

This expression for a computed field in the crosstab returns the average of the employee counts (the first expression):

```
CrosstabAvg(1)
```

This expression for a computed field in the crosstab returns the average of the salary totals (the second expression):

```
CrosstabAvg(2)
```

Consider a crosstab that has two columns (region and city) and the values expression Avg(sales for crosstab). This expression for a computed field in the detail band computes the average sales over all the cities in a region:

```
CrosstabAvg(1, 2, "@region")
```

This expression for another computed field in the same crosstab computes the grand average over all the cities:

```
CrosstabAvg(1)
```

### See also

CrosstabCount  
CrosstabMax  
CrosstabMin  
CrosstabSum

## CrosstabCount

**Description** Counts the number of values returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabCount can also count the number of the expression's values for groups of column values.

---

### For crosstabs only

You can use this function *only* in a crosstab DataWindow object or report.

---

### Syntax

**CrosstabCount** ( *n* { *column*, *groupvalue* } )

Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the total number of returned values
<i>column</i> (optional)	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog for which you want intermediate calculations
<i>groupvalue</i> (optional)	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name as a quoted string

### Return value

Long. Returns the number of values returned by expression *n* for all the column values or, optionally, for a subset of column values.

### Usage

This function is meaningful *only* for the count of the values of the expression in a *row* in the crosstab. This means you can use it only in the detail band, not in a header, trailer, or summary band.

NULL values are ignored and are not included in the count.

**FOR INFO** For more information about restricting the calculation to groups of values when the crosstab definition has more than one column, see Usage for CrosstabAvg.

---

### Reviewing the expressions

To review the expressions defined for the crosstab values, open the Crosstab Definition dialog box (display the popup menu in an unused area of the crosstab and then select Crosstab).

---

### Examples

These examples all use the crosstab-values expressions shown below:

```
Count(emp_id for crosstab),Sum(salary for crosstab)
```

This expression for a computed field in the crosstab returns the count of the employee counts (the first expression):

```
CrosstabCount (1)
```

This expression for a computed field in the crosstab returns the count of the salary totals (the second expression):

```
CrosstabCount (2)
```

The next two examples use a crosstab with two columns (year and quarter), a row (product), and the values expression Avg(sales for crosstab).

This expression for a computed field returns the count of the sales for each year:

```
CrosstabCount (1, 2, "@year")
```

This expression for a computed field returns the count of all the sales in the row:

```
CrosstabCount (1)
```

For an example illustrating how the DataWindow painter and Report painter automatically define a crosstab by creating computed fields using the Crosstab functions, see CrosstabAvg.

### See also

CrosstabAvg  
CrosstabMax  
CrosstabMin  
CrosstabSum

## CrosstabMax

**Description** Calculates the maximum value returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabMax can also calculate the maximum of the expression's values for groups of column values.

---

### For crosstabs only

You can use this function *only* in a crosstab DataWindow object or report.

---

### Syntax

**CrosstabMax** ( *n* { *column*, *groupvalue* } )

Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the maximum returned value. The expression's data type must be numeric
<i>column</i> (optional)	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations
<i>groupvalue</i> (optional)	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name as a quoted string

**Return value** Double. Returns the maximum value returned by expression *n* for all the column values or, optionally, for a subset of column values.

**Usage** This function is meaningful *only* for the maximum of the values of the expression in a *row* in the crosstab. This means you can use it only in the detail band, not in a header, trailer, or summary band.

NULL values are ignored and are not included in the comparison.

**FOR INFO** For more information about restricting the calculation to groups of values when the crosstab definition has more than one column, see Usage for CrosstabAvg.

---

### Reviewing the expressions

To review the expressions defined for the crosstab values, open the Crosstab Definition dialog box (display the popup menu in an unused area of the crosstab and then select Crosstab).

---

### Examples

These examples all use the crosstab-values expressions shown below:

```
Count(emp_id for crosstab),Sum(salary for crosstab)
```

This expression for a computed field in the crosstab returns the maximum of the employee counts (the first expression):

```
CrosstabMax(1)
```

This expression for a computed field in the crosstab returns the maximum of the salary totals (the second expression):

```
CrosstabMax(2)
```

The next two examples use a crosstab with two columns (year and quarter), a row (product), and a values expression Avg(sales for crosstab).

This expression for a computed field returns the largest of the quarterly average sales for each year:

```
CrosstabMax(1, 2, "@year")
```

This expression for a computed field returns the maximum of all the average sales in the row:

```
CrosstabMax(1)
```

For an example illustrating how the DataWindow painter and Report painter automatically define a crosstab by creating computed fields using the Crosstab functions, see CrosstabAvg.

### See also

CrosstabAvg  
CrosstabCount  
CrosstabMin  
CrosstabSum

## CrosstabMin

**Description** Calculates the minimum value returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabMin can also calculate the minimum of the expression's values for groups of column values.

---

### For crosstabs only

You can use this function *only* in a crosstab DataWindow object or report.

---

### Syntax

**CrosstabMin** ( *n*{, *column*, *groupvalue* } )

Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the minimum return value. The expression's data type must be numeric
<i>column</i> (optional)	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations
<i>groupvalue</i> (optional)	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name as a quoted string

**Return value** Double. Returns the minimum value returned by expression *n* for all the column values or, optionally, for a subset of column values.

**Usage** This function is meaningful *only* for the minimum of the values of the expression in a *row* in the crosstab. This means you can use it only in the detail band, not in a header, trailer, or summary band.

NULL values are ignored and are not included in the comparison.

**FOR INFO** For more information about restricting the calculation to groups of values when the crosstab definition has more than one column, see Usage for CrosstabAvg.

---

### Reviewing the expressions

To review the expressions defined for the crosstab values, open the Crosstab Definition dialog box (display the popup menu in an unused area of the crosstab and then select Crosstab).

---

Examples

These examples all use the crosstab-values expressions shown below:

```
Count(emp_id for crosstab),Sum(salary for crosstab)
```

This expression for a computed field in the crosstab returns the minimum of the employee counts (the first expression):

```
CrosstabMin(1)
```

This expression for a computed field in the crosstab returns the minimum of the salary totals (the second expression):

```
CrosstabMin(2)
```

The next two examples use a crosstab with two columns (year and quarter), a row (product), and the values expression Avg(sales for crosstab).

This expression for a computed field returns the smallest of the quarterly average sales for each year:

```
CrosstabMin(1, 2, "@year")
```

This expression for a computed field returns the minimum of all the average sales in the row:

```
CrosstabMin(1)
```

For an example illustrating how the DataWindow painter and Report painter automatically define a crosstab by creating computed fields using the crosstab functions, see CrosstabAvg.

See also

CrosstabAvg  
CrosstabCount  
CrosstabMax  
CrosstabSum



## CrosstabSum

**Description** Calculates the sum of the values returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabSum can also calculate the sum of the expression's values for groups of column values.

---

### For crosstabs only

You can use this function *only* in a crosstab DataWindow object or report.

---

**Syntax**

**CrosstabSum** ( *n* {, *column*, *groupvalue* } )

Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the sum of the returned values. The expression's data type must be numeric
<i>column</i> (optional)	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations
<i>groupvalue</i> (optional)	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name as a quoted string

**Return value**

Double. Returns the total of the values returned by expression *n* for all the column values or, optionally, for a subset of column values.

**Usage**

This function is meaningful *only* for the sum of the values of the expression in a *row* in the crosstab. This means you can use it only in the detail band, not in a header, trailer, or summary band.

NULL values are ignored and are not included in the sum.

**FOR INFO** For more information about restricting the calculation to groups of values when the crosstab definition has more than one column, see Usage for CrosstabAvg.

---

### Reviewing the expressions

To review the expressions defined for the crosstab values, open the Crosstab Definition dialog box (display the popup menu in an unused area of the crosstab and then select Crosstab).

---

Examples

These examples all use the crosstab-values expressions shown below:

```
Count(emp_id for crosstab),Sum(salary for crosstab)
```

This expression for a computed field in the crosstab returns the sum of the employee counts (the first expression):

```
CrosstabSum(1)
```

This expression for a computed field in the crosstab returns the sum of the salary totals (the second expression):

```
CrosstabSum(2)
```

The next two examples use a crosstab with two columns (year and quarter), a row (product), and the values expression Avg(sales for crosstab).

This expression for a computed field returns the sum of the quarterly average sales for each year:

```
CrosstabSum(1, 2, "@year")
```

This expression for a computed field returns the sum of all the average sales in the row:

```
CrosstabSum(1)
```

For an example illustrating how the DataWindow painter and Report painter automatically define a crosstab by creating computed fields using the Crosstab functions, see CrosstabAvg.

See also

CrosstabAvg  
CrosstabCount  
CrosstabMax  
CrosstabMin

## CumulativePercent

**Description** Calculates the total value of the rows up to and including the current row in the specified column as a percentage of the total value of the column (a running percentage).

**Syntax** **CumulativePercent** ( *column* { FOR *range* } )

Argument	Description
<i>column</i>	The column for which you want the cumulative value of the rows up to and including the current row as a percentage of the total value of the column for <i>range</i> . <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the cumulative percentage. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The cumulative percentage of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The cumulative percentage of rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The cumulative percentage of the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The cumulative percentage of all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The cumulative percentage of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The cumulative percentage of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>

**Return value** Long. Returns the cumulative percentage value.

**Usage** If you specify *range*, CumulativePercent restarts the accumulation at the start of the range.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

In calculating the percentage, NULL values are ignored.

---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or a report always retrieves all rows.

**Examples**

This expression returns the running percentage for the values that are not NULL in the column named salary:

```
CumulativePercent(salary)
```

This expression returns the running percentage for the column named salary for the values in group 1 that are not NULL:

```
CumulativePercent(salary for group 1)
```

This expression entered in the Value box on the Data property page in the Graph Object property sheet returns the running percentage for the salary column for the values in the graph that are not NULL:

```
CumulativePercent(salary for graph)
```

This expression in a crosstab computed field returns the running percentage for the salary column for the values in the crosstab that are not NULL:

```
CumulativePercent(salary for crosstab)
```

**See also**

Percent  
CumulativeSum

# CumulativeSum

**Description** Calculates the total value of the rows up to and including the current row in the specified column (a running total).

**Syntax** **CumulativeSum** ( *column* { FOR *range* } )

Argument	Description
<i>column</i>	The column for which you want the cumulative total value of the rows up to and including the current row for group. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the cumulative sum. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The cumulative sum of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The cumulative sum of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The cumulative sum of the values in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The cumulative sum of all values in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The cumulative sum of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The cumulative sum of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>

**Return value** Long. Returns the cumulative total value of the rows.

**Usage** If you specify *range*, CumulativeSum restarts the accumulation at the start of the range.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

In calculating the sum, NULL values are ignored.

#### Examples

This expression returns the running total for the values that are not NULL in the column named salary:

```
CumulativeSum(salary)
```

This expression returns the running total for the values that are not NULL in the column named salary in group 1:

```
CumulativeSum(salary for group 1)
```

This expression returns the running total for the values that are not NULL in the column named salary in group 1:

```
CumulativeSum(salary for group 1)
```

This expression entered in the Value box on the Data property page in the Graph Object property sheet returns the running total for the salary column for the values in the graph that are not NULL:

```
CumulativeSum(salary for graph)
```

This expression in a crosstab computed field returns the running total for the salary column for the values in the crosstab that are not NULL:

```
CumulativeSum(salary for crosstab)
```

#### See also

CumulativePercent

## CurrentRow

Description	Reports the number of the current row (the row with focus) in a DataWindow object or form.
Syntax	<b>CurrentRow</b> ( )
Return value	Long. Returns the number of the row if it succeeds and 0 if no row is current.

---

### What row is current

The current row is not always a row displayed on the screen. For example, if the cursor is on row 7 column 2 and the user uses the scroll bar to scroll to row 50, the current row remains row 7 unless the user clicks row 50.

---

**Examples** This expression in a computed field returns the number of the current row:

```
CurrentRow( )
```

This expression for a computed object displays an arrow bitmap as an indicator for the row with focus and displays no bitmap for rows not having focus. As the user moves from row to row in the DataWindow object or form, an arrow marks where the user is:

```
Bitmap(If(CurrentRow( ) = GetRow(), "arrow.bmp", ""))
```

Alternatively, this expression for the Visible property of an arrow picture object makes the arrow bitmap visible for the row with focus and invisible for rows not having focus. As the user moves from row to row in the DataWindow object or form, an arrow marks where the user is:

```
If(CurrentRow( ) = GetRow(), 1, 0)
```

**See also** [GetRow](#)

## Date

**Description** Converts a string whose value is a valid date to a value of data type date.

**Syntax** **Date** ( *string* )

<b>Argument</b>	<b>Description</b>
<i>string</i>	A string containing a valid date (such as Jan 1, 1998, or 12-31-99) that you want returned as a date

**Return value** Date. Returns the date in *string* as a date. If *string* does not contain a valid date, Date returns NULL.

**Usage** The value of the string must be a valid date.

---

### Valid dates

Valid dates can include any combination of day (1–31), month (1–12 or the name or abbreviation of a month), and year (two or four digits). Leading zeros are optional for month and day. If the month is a name or an abbreviation, it can come before or after the day; if it is a number, it must be in the month location specified in the Windows control panel. A 4-digit number is assumed to be a year.

---

An expression has a more limited set of data types than the functions that can be part of the expression. Although the Date function returns a date value, the whole expression is promoted to a DateTime value. Therefore, if your expression consists of a single Date function, it will appear that Date returns the wrong data type. To display the date without the time, choose an appropriate display format. (See "Using DataWindow painter and InfoMaker functions" on page 706.)

**Examples** These expressions all return the date data type for July 4, 1994 (1994-07-04), when the default position of month is center:

```
Date ("1994/07/04")  
Date ("1994 July 4")  
Date ("July 4, 1994")
```

**See also** IsDate



## DateTime

**Description** Combines a date and a time value into a DateTime value.

**Syntax** **DateTime** ( *date* {, *time* } )

Argument	Description
<i>date</i>	A valid date (such as Jan 1, 1998, or 12-31-99) or a blob variable whose first value is a date that you want included in the value returned by DateTime
<i>time</i> (optional)	A valid time (such as 8am or 10:25:23:456799) or a blob variable whose first value is a time that you want included in the value returned by DateTime. If you include a time, only the hour portion is required. If you omit the minutes, seconds, or microseconds, they are assumed to be zeros. If you omit am or pm, the hour is determined according to the 24-hour clock

**Return value** DateTime. Returns a DateTime value based on the values in *date* and optionally *time*. If time is omitted, DateTime uses 00:00:00.000000 (midnight).

**Usage** To display microseconds in a time, the display format for the field must include microseconds.

FOR INFO For information on valid dates, see Date.

**Examples** This expression returns the values in the order\_date and order\_time columns as a DateTime value that can be used to update the database:

```
DateTime(Order_Date, Order_Time)
```

**See also** Date  
Time

## Day

**Description** Obtains the day of the month in a date value.

**Syntax** **Day** ( *date* )

<b>Argument</b>	<b>Description</b>
<i>date</i>	The date from which you want the day

**Return value** Integer. Returns an integer (1–31) representing the day of the month in *date*.

**Examples** This expression returns 31:

**Day** (1994-01-31)

This expression returns the day of the month in the *start\_date* column:

**Day** (start\_date)

**See also** Date  
IsDate  
Month  
Year

# DayName

**Description** Gets the day of the week in a date value and returns the weekday's name.

**Syntax** **DayName** ( *date* )

Argument	Description
<i>date</i>	The date for which you want the name of the day

**Return value** String. Returns a string whose value is the name of the weekday (Sunday, Monday, and so on) for *date*.

**Usage** DayName returns a name in the language of the deployment kit (DDDK) available on the machine where the application is run. If you have installed a localized deployment kit (DDDK) in the development environment or on a user's machine, then on that machine the name returned by DayName will be in the language of the localized DDDK.

**FOR INFO** For information about localized deployment kits, which are available in French, German, Italian, Spanish, Dutch, Danish, Norwegian, and Swedish, see "The deployed application" on page 649.

**Examples** This expression for a computed field returns Okay if the day in *date\_signed* is not Sunday:

```
If (DayName (date_signed) <> "Sunday", "Okay", "Invalid Date")
```

To pass this validation rule, the day in *date\_signed* must not be Sunday:

```
DayName (date_signed) <> "Sunday"
```

**See also** Date  
Day  
DayNumber  
IsDate

# DayNumber

**Description** Gets the day of the week of a date value and returns the number of the weekday.

**Syntax** **DayNumber** ( *date* )

<b>Argument</b>	<b>Description</b>
<i>date</i>	The date from which you want the number of the day of the week

**Return value** Integer. Returns an integer (1–7) representing the day of the week of *date*. Sunday is day 1, Monday is day 2, and so on.

**Examples** This expression for a computed field returns Wrong Day if the date in start\_date is not a Sunday or a Monday:

```
If(DayNumber(start_date) > 2, "Okay", "Wrong Day")
```

This expression for a computed field returns Wrong Day if the date in end\_date is a Saturday or a Sunday:

```
If(DayNumber(end_date) > 1 and DayNumber(end_date) < 7, "Okay", "Wrong Day")
```

This validation rule for the column end\_date ensures that the day is not a Saturday or Sunday:

```
DayNumber(end_date) >1 and DayNumber(end_date) < 7
```

**See also**

Date  
Day  
DayName  
IsDate

## DaysAfter

**Description** Gets the number of days one date occurs after another.

**Syntax** **DaysAfter** ( *date1*, *date2* )

Argument	Description
<i>date1</i>	A date value that is the start date of the interval being measured
<i>date2</i>	A date value that is the end date of the interval

**Return value** Long. Returns a long containing the number of days *date2* occurs after *date1*. If *date2* occurs before *date1*, DaysAfter returns a negative number.

**Examples** This expression returns 4:

```
DaysAfter(1995-12-20, 1995-12-24)
```

This expression returns -4:

```
DaysAfter(1995-12-24, 1995-12-20)
```

This expression returns 0:

```
DaysAfter(1995-12-24, 1995-12-24)
```

This expression returns 5:

```
DaysAfter(1994-12-29, 1995-01-03)
```

**See also**

Date  
SecondsAfter

## Describe

**Description** Reports the values of attributes of a report or form object and objects within it. Each column and graphic object in the report or form has a set of attributes. You specify one or more properties as a string and Describe returns the values of the attributes.

**Syntax** **Describe** ( *propertylist* )

<b>Argument</b>	<b>Description</b>
<i>propertylist</i>	A string whose value is a blank-separated list of properties or Evaluate functions.

**Return value** String. Returns a string that includes a value for each property or Evaluate function. A newline character (~n) separates the value of each item in *propertylist*.  
If *propertylist* contains an invalid item, Describe returns an exclamation point (!) for that item and ignores the rest of *propertylist*. Describe returns a question mark (?) if there is no value for a property.

**Usage** The values for *propertylist* can be complex. For complete information and examples, see the Describe function in the *PowerScript Reference* in the PowerBuilder documentation set.

**Examples** This expression for a computed field in the header band of a DataWindow object displays the DataWindow object's SELECT statement:

```
Describe ("DataWindow.Table.Select")
```

## Exp

Description Raises  $e$  to the specified power.

Syntax **Exp** (  $n$  )

Argument	Description
$n$	The power to which you want to raise $e$ (2.71828)

Return value Double. Returns  $e$  raised to the power  $n$ .

Examples This expression returns 7.38905609893065:

**Exp** ( 2 )

See also  
Log  
LogTen

## Fact

Description Gets the factorial of a number.

Syntax **Fact** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The number for which you want the factorial

Return value Double. Returns the factorial of *n*.

Examples This expression returns 24:

**Fact** ( 4 )

Both these expressions return 1:

**Fact** ( 1 )

**Fact** ( 0 )



## Fill

Description	Builds a string of the specified length by repeating the specified characters until the result string is long enough.						
Syntax	<p><b>Fill</b> ( <i>chars</i>, <i>n</i> )</p> <table border="1"> <thead> <tr> <th>Argument</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>chars</i></td> <td>A string whose value will be repeated to fill the return string</td> </tr> <tr> <td><i>n</i></td> <td>A long whose value is the length of the string you want returned</td> </tr> </tbody> </table>	Argument	Description	<i>chars</i>	A string whose value will be repeated to fill the return string	<i>n</i>	A long whose value is the length of the string you want returned
Argument	Description						
<i>chars</i>	A string whose value will be repeated to fill the return string						
<i>n</i>	A long whose value is the length of the string you want returned						
Return value	String. Returns a string <i>n</i> characters long filled with repetitions of the characters in the argument <i>chars</i> . If the argument <i>chars</i> has more than <i>n</i> characters, the first <i>n</i> characters of <i>chars</i> are used to fill the return string. If the argument <i>chars</i> has fewer than <i>n</i> characters, the characters in <i>chars</i> are repeated until the return string has <i>n</i> characters.						
Usage	Fill is used to create a line or other special effect. For example, asterisks repeated in a printed report can fill an amount line, or hyphens can simulate a total line in a screen display.						
Examples	<p>This expression returns a string containing 35 stars:</p> <pre>Fill (" * ", 35)</pre> <p>This expression returns the string -+--+--:</p> <pre>Fill (" -+", 7)</pre> <p>This expression returns 10 tildes (~):</p> <pre>Fill (" ~", 10)</pre>						
See also	Space						

# First

Description

Reports the value in the first row in the specified column.

Syntax

**First** ( *column* { FOR *range* { DISTINCT { *expressn* {, *express2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the value of the first row. <i>Column</i> can be a column name or a column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column
FOR <i>range</i> (optional)	<p>The data that will be included when the value in the first row is found. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The value in the first of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The value in the first of rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The value in the first of the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The value in the first of all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The value in the first row in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The value in the first row in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes First to consider only the distinct values in <i>column</i> when determining the first value. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expressn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expressn</i> can be the name of a column, a function, or an expression

Return value

The data type of the column. Returns the value in the first row of *column*. If you specify *range*, First returns the value of the first row in *column* in *range*.

## Usage

If you specify *range*, First determines the value of the first row in *column* in *range*. If you specify DISTINCT, First returns the first distinct value in *column*, or if you specify *expressn*, the first distinct value in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or a report always retrieves all rows.

## Examples

This expression returns the first value in column 3 on the page:

```
First(#3 for page)
```

This expression returns the first distinct value in the column named dept\_id in group 2:

```
First(dept_id for group 2 DISTINCT)
```

This expression returns the first value in the column named dept\_id in group 2:

```
First(dept_id for group 2)
```

## See also

Last

# GetRow

**Description** Reports the number of a row associated with a band in a DataWindow object or a report.

**Syntax** **GetRow ( )**

**Return value** Long. Returns the number of a row if it succeeds, 0 if no data has been retrieved or added, and -1 if an error occurs. Where you call GetRow determines what row it returns, as follows:

<b>If the object in the DataWindow object or report is in this band</b>	<b>GetRow returns</b>
Header	First row on the page
Group header	First row in the group
Detail	The row in which the expression occurs
Group trailer	Last row in the group
Summary	Last row in the report or DataWindow object
Footer	Last row on the page

**Examples** This expression for a computed field in the detail band displays the number of each row:

```
GetRow()
```

This expression for a computed field in the header band checks to see if there is data. It returns the number of the first row on the page if there is data, and otherwise returns No Data:

```
If(GetRow()= 0, "No Data", String(GetRow()))
```

**See also** CurrentRow

## GetText

Description	Obtains the text that a user has entered in a column in a DataWindow object or form.
Syntax	<b>GetText ( )</b>
Return value	String. Returns the text the user has entered in the current column.
Usage	Use <code>GetText</code> in validation rules to compare what the user has entered to application-defined criteria before it is accepted into the data buffer.
Examples	This validation rule checks that the value the user entered in the column is less than 100:

```
Integer(GetText()) < 100
```

# Hour

**Description** Obtains the hour in a time value. The hour is based on a 24-hour clock.

**Syntax** **Hour** ( *time* )

<b>Argument</b>	<b>Description</b>
<i>time</i>	The time value from which you want the hour

**Return value** Integer. Returns an integer (00–23) containing the hour portion of *time*.

**Examples** This expression returns the current hour:

**Hour** (Now ( ) )

This expression returns 19:

**Hour** (19:01:31)

**See also** Minute  
Now  
Second

**If**

**Description** Evaluates a condition and returns a value based on that condition.

**Syntax** `If ( boolean, truevalue, falsevalue )`

Argument	Description
<i>boolean</i>	A boolean expression that evaluates to TRUE or FALSE
<i>truevalue</i>	A string containing the value you want returned if the boolean expression is TRUE
<i>falsevalue</i>	A string containing the value you want returned if the boolean expression is FALSE

**Return value** The data type of *truevalue* or *falsevalue*. Returns *truevalue* if *boolean* is TRUE and *falsevalue* if it is FALSE. Returns NULL if an error occurs.

**Examples** This expression returns Boss if salary is over \$100,000 and Employee if salary is less than or equal to \$100,000:

```
If(salary > 100000, "Boss", "Employee")
```

This expression returns Boss if salary is over \$100,000, Supervisor if salary is between \$12,000 and \$100,000, and Clerk if salary is less than or equal to \$12,000:

```
If(salary > 100000, "Boss", If(salary > 12000,
"Supervisor", "Clerk"))
```

In this example of a validation rule, the value the user should enter in the commission column depends on the price. If price is greater than or equal to 1000, then the commission is between .10 and .20. If price is less than 1000, then the commission must be between .04 and .09. The validation rule is:

```
(Number(GetText()) >= If(price >=1000, .10, .04)) AND
(Number(GetText()) <= If(price >= 1000, .20, .09))
```

The accompanying error message expression might be:

```
"Price is " + If(price >= 1000, "greater than or
equal to", "less than") + " 1000. Commission must be
between " + If(price >= 1000, ".10", ".04") + " and
" + If(price >= 1000, ".20.", ".09.")
```

See also

Case

# Int

**Description** Gets the largest whole number less than or equal to a number.

**Syntax** `Int ( n )`

<b>Argument</b>	<b>Description</b>
<i>n</i>	The number for which you want the largest whole number that is less than or equal to it

**Return value** The data type of *n*. Returns the largest whole number less than or equal to *n*.

**Examples** These expressions return 3.0:

`Int ( 3 . 2 )`

`Int ( 3 . 8 )`

These expressions return -4.0:

`Int ( -3 . 2 )`

`Int ( -3 . 8 )`

**See also** Ceiling  
Integer  
Round  
Truncate



# Integer

**Description** Converts the value of a string to an integer.

**Syntax** **Integer** ( *string* )

Argument	Description
<i>string</i>	The string you want returned as an integer

**Return value** Integer. Returns the contents of *string* as an integer if it succeeds and 0 if *string* is not a number.

**Examples** This expression converts the string 24 to an integer:

```
Integer ("24")
```

This expression for a computed field returns "Not a valid age" if age does not contain a number. The expression checks whether the Integer function returns 0, which means it failed to convert the value:

```
If (Integer(age) <> 0, age, "Not a valid age")
```

This expression returns 0:

```
Integer("3ABC") // 3ABC is not a number
```

This validation rule checks that the value in the column the user entered is less than 100:

```
Integer(GetText()) < 100
```

This validation rule for the column named age insures that age contains a string:

```
Integer(age) <> 0
```

**See also** IsNumber

## IsDate

Description Tests whether a string value is a valid date.

Syntax **IsDate** ( *datevalue* )

Argument	Description
<i>datevalue</i>	A string whose value you want to test to determine whether it is a valid date

Return value Boolean. Returns TRUE if *datevalue* is a valid date and FALSE if it is not.

Examples This expression returns TRUE:

```
IsDate ("Jan 1, 95")
```

This expression returns FALSE:

```
IsDate ("Jan 32, 1997")
```

This expression for a computed field is entered in the Computed Object property sheet. It returns the number of the day in `date_received` in the computed field if the column contains a valid date, and otherwise returns 0:

```
If(IsDate(String(date_received)),  
DayNumber(date_received), 0)
```

## IsNull

**Description** Reports whether the value of a column or expression is NULL.

**Syntax** **IsNull** ( *any* )

Argument	Description
<i>any</i>	A column or expression that you want to test to determine whether its value is NULL

**Return value** Boolean. Returns TRUE if *any* is NULL and FALSE if it is not.

**Usage** Use IsNull to test whether a user-entered value or a value retrieved from the database is NULL.

**Examples** This expression returns TRUE if either a or b is NULL:

```
IsNull ( a + b )
```

This expression returns TRUE if the value in the salary column is NULL:

```
IsNull ( salary )
```

This expression returns TRUE if the value the user has entered is NULL:

```
IsNull ( GetText ( ) )
```

# IsNumber

Description Reports whether the value of a string is a number.

Syntax **IsNumber** ( *string* )

Argument	Description
<i>string</i>	A string whose value you want to test to determine whether it is a valid number

Return value Boolean. Returns TRUE if *string* is a valid number and FALSE if it is not.

Examples This expression returns TRUE:

```
IsNumber ("32.65")
```

This expression returns FALSE:

```
IsNumber ("A16")
```

This expression for a computed field returns "Not a valid age" if age does not contain a number:

```
If(IsNumber(age), age, "Not a valid age")
```

To pass this validation rule, Age\_nbr must be a number:

```
IsNumber(Age_nbr) = TRUE
```

See also Integer

## IsRowModified

Description	Reports whether the row has been modified in the DataWindow object or form.
Syntax	<b>IsRowModified</b> ( )
Return value	Boolean. Returns TRUE if the row has been modified and FALSE if it has not.
Usage	In a DataWindow object, when you use IsRowModified in bands other than the detail band, it reports on a row in the detail band. See GetRow for a table specifying which row is associated with each band for reporting purposes.
Examples	<p>This expression in a computed field in the detail area displays TRUE or FALSE to indicate whether each row has been modified:</p> <pre>IsRowModified()</pre> <p>This expression for the Color property on the Expressions property page in the Column Object property sheet displays the associated column text in red if the user has modified any value in the row:</p> <pre>If(IsRowModified(), 255, 0)</pre>
See also	GetRow

## IsRowNew

Description	Reports whether the row has been newly inserted in the DataWindow object or form.
Syntax	<b>IsRowNew</b> ( )
Return value	Boolean. Returns TRUE if the row is new and FALSE if it was retrieved from the database.
Usage	In a DataWindow object, when you call IsRowNew in bands other than the detail band, it reports on a row in the detail band. See GetRow for a table specifying which row is associated with each band for reporting purposes.
Examples	This expression for the Protect property on the Expressions property page in the Column Object property sheet prevents the user from modifying the associated column unless the row has been newly inserted:  <code>If(IsRowNew(), 0, 1)</code>
See also	GetRow

## IsSelected

Description	Determines whether the row is selected. A selected row is highlighted using reverse video.
Syntax	<b>IsSelected</b> ( )
Return value	Boolean. Returns TRUE if the row is selected and FALSE if it is not selected.
Usage	When you use IsSelected in bands other than the detail band, it reports on a row in the detail band. See GetRow for a table specifying which row is associated with each band for reporting purposes.
Examples	<p>This expression for a computed field in the detail area displays a bitmap if the row is selected:</p> <pre>Bitmap(If(IsSelected(), "beach.bmp", ""))</pre> <p>This example allows the DataWindow object to display a salary total for all the selected rows. The expression for a computed field in the detail band returns the salary only when the row is selected so that another computed field in the summary band can add up all the selected salaries.</p> <p>The expression for cf_selected_salary (the computed field in the detail band) is:</p> <pre>If(IsSelected(), salary, 0)</pre> <p>The expression for the computed field in the summary band is:</p> <pre>Sum(cf_selected_salary for all)</pre>
See also	GetRow

## IsTime

Description

Reports whether the value of a string is a valid time value.

Syntax

**IsTime** ( *timevalue* )

<b>Argument</b>	<b>Description</b>
<i>timevalue</i>	A string whose value you want to test to determine whether it is a valid time

Return value

Boolean. Returns TRUE if *timevalue* is a valid time and FALSE if it is not.

Examples

This expression returns TRUE:

```
IsTime ("8:00:00 am")
```

This expression returns FALSE:

```
IsTime ("25:00")
```

To pass this validation rule, the value in `start_time` must be a time:

```
IsTime (start_time)
```



# Large

## Description

Finds a large value at a specified ranking in a column (for example, third-largest, fifth-largest) and returns the value of another column or expression based on the result.

## Syntax

**Large** ( *returnexp*, *column*, *ntop* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>returnexp</i>	The value you want returned when the large value is found. <i>Returnexp</i> includes a reference to a column, but not necessarily the column that is being evaluated for the largest value, so that a value is returned from the same row that contains the large value
<i>column</i>	The column that contains the large value you are searching for. <i>Column</i> can be a column name or a column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
<i>ntop</i>	The ranking of the large value in relation to the column's largest value. For example, when <i>ntop</i> is 2, Large finds the second-largest value
FOR <i>range</i> (optional)	<p>The data that will be included when the largest value is found. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The largest of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The largest of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The largest of the values in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The largest of all values in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The largest of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The largest of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>

Argument	Description
DISTINCT (optional)	Causes Large to consider only the distinct values in <i>column</i> when determining the large value. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expressn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expressn</i> can be the name of a column, a function, or an expression

Return value                    The data type of *returnexp*. Returns the *ntop*-largest value if it succeeds and -1 if an error occurs.

Usage

If you specify *range*, Large returns the value in *returnexp* when the value in *column* is the *ntop*-largest value in *range*. If you specify DISTINCT, Large returns *returnexp* when the value in *column* is the *ntop*-largest value of the distinct values in *column*, or if you specify *expressn*, the *ntop*-largest for each distinct value of *expressn*.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All
- ◆ For Graph objects, Rows can be All, Page, or Group
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies

---

**Max may be faster**

If you don't need a return value from another column and you want to find the largest value (*ntop* = 1), use Max; it is faster.

---



---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

**Examples**

These expressions return the names of the salespersons with the three largest sales (sum\_sales is the sum of the sales for each salesperson) in group 2, which might be the salesregion group. Note that sum\_sales contains the values being compared, but Large returns a value in the name column:

```
Large(name, sum_sales, 1 for group 2)
```

```
Large(name, sum_sales, 2 for group 2)
```

```
Large(name, sum_sales, 3 for group 2)
```

This example reports the salesperson with the third-largest sales, considering only the first entry for each person:

```
Large(name, sum_sales, 3 for all DISTINCT sum_sales)
```

**See also**

**Small**

# Last

Description

Gets the value in the last row in the specified column.

Syntax

**Last** ( *column* { FOR *range* { DISTINCTT { *expres1* {, *expres2* {, ... } } } } } }

Argument	Description
<i>column</i>	The column for which you want the value of the last row. <i>Column</i> can be a column name or a column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column.
FOR <i>range</i> (optional)	The data that will be included when the value in the last row is found. For most presentation styles, values for <i>range</i> are: <ul style="list-style-type: none"> <li>◆ ALL — (Default) The value in the last of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The value in the last row in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The value in the last row in <i>column</i> on a page</li> </ul> For Crosstabs, specify CROSSTAB for <i>range</i> : <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The value in the last row in <i>column</i> in the crosstab</li> </ul> For Graph and OLE objects, specify the type of object for <i>range</i> . The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are: <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The value in the last row in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The value in the last row in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Last to consider only the distinct values in <i>column</i> when determining the last value. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Return value

The data type of the column. Returns the value in the last row of *column*. If you specify *range*, Last returns the value of the last row in *column* in *range*.

## Usage

If you specify *range*, Last determines the value of the last row in *column* in *range*. If you specify DISTINCT, Last returns the last distinct value in *column*, or if you specify *expressn*, the last distinct value in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

## Examples

This expression returns the last distinct value in the column named dept\_id in group 2:

```
Last(dept_id for group 2 DISTINCT)
```

This expression returns the last value in the column named emp\_id in group 2:

```
Last(emp_id for group 2)
```

## See also

First

## Left

**Description** Obtains a specified number of characters from the beginning of a string.

**Syntax** **Left** (*string*, *n*)

<b>Argument</b>	<b>Description</b>
<i>string</i>	The string containing the characters you want
<i>n</i>	A long specifying the number of characters you want

**Return value** String. Returns the leftmost *n* characters in *string* if it succeeds and the empty string ("") if an error occurs.

If *n* is greater than or equal to the length of the string, Left returns the entire string. It does not add spaces to make the return value's length equal to *n*.

**Examples** This expression returns BABE:

```
Left ("BABE RUTH", 4)
```

This expression returns BABE RUTH:

```
Left ("BABE RUTH", 40)
```

This expression for a computed field returns the first 40 characters of the text in the column `home_address`:

```
Left(home_address, 40)
```

**See also** Mid  
Pos  
Right

## LeftTrim

**Description** Removes spaces from the beginning of a string.

**Syntax** **LeftTrim** ( *string* )

Argument	Description
<i>string</i>	The string you want returned with leading spaces deleted

**Return value** String. Returns a copy of *string* with leading spaces deleted if it succeeds and the empty string ("") if an error occurs.

**Examples** This expression returns RUTH:

```
LeftTrim( " RUTH" )
```

This expression for a computed field deletes any leading blanks from the value in the column *lname* and returns the value preceded by the salutation specified in *salut\_emp*:

```
salut_emp + " " + LeftTrim(lname)
```

**See also** RightTrim  
Trim

## Len

Description Reports the length of a string.

Syntax **Len** ( *string* )

Argument	Description
<i>string</i>	The string for which you want the length

Return value Long. Returns a long containing the length of *string* if it succeeds and -1 if an error occurs.

Examples This expression returns 0:

```
Len ( " " )
```

This validation rule tests that the value the user entered is fewer than 20 characters:

```
Len (GetText ()) < 20
```



# Log

Description Gets the natural logarithm of a number.

Syntax

**Log** ( *n* )

Argument	Description
<i>n</i>	The number for which you want the natural logarithm (base e). The value of <i>n</i> must be greater than 0

Return value

Double. Returns the natural logarithm of *n*. An execution error occurs if *n* is negative or zero.

---

### Inverse

The inverse of the Log function is the Exp function.

---

Examples

This expression returns 2.302585092:

**Log** (10)

This expression returns  $-.693147 \dots$ :

**Log** (0.5)

Both these expressions result in an error during execution:

**Log** (0)

**Log** (-2)

See also

Exp  
LogTen

# LogTen

Description Gets the base 10 logarithm of a number.

Syntax **LogTen** ( *n* )

Argument	Description
<i>n</i>	The number for which you want the base 10 logarithm. The value of <i>n</i> must not be negative

Return value Double. Returns the base 10 logarithm.

---

**Obtaining a number**

The expression  $10^n$  is the inverse for  $\text{LogTen}(n)$ . To obtain *n* given number ( $\text{nbr} = \text{LogTen}(n)$ ), use  $n = 10^{\text{nbr}}$ .

---

Examples This expression returns 1:

**LogTen** (10)

The following expressions both return 0:

**LogTen** (1)

**LogTen** (0)

This expression results in an execution error:

**LogTen** (-2)

See also Log

## Long

Description Converts the value of a string to a long.

Syntax **Long** ( *string* )

<b>Argument</b>	<b>Description</b>
<i>string</i>	The string you want returned as a long

Return value Long. Returns the contents of *string* as a long if it succeeds and 0 if *string* is not a valid number.

Examples This expression returns 2167899876 as a long:

```
Long ( "2167899876" )
```

## LookUpDisplay

**Description** Obtains the display value in the code table associated with the data value in the specified column.

**Syntax** **LookUpDisplay** ( *column* )

<b>Argument</b>	<b>Description</b>
<i>column</i>	The column for which you want the code table display value

**Return value** String. Returns the display value when it succeeds and the empty string ("") if an error occurs.

**Usage** If a column has a code table, a buffer stores a value from the data column of the code table, but the user sees a value from the display column. Use LookUpDisplay to get the value the user sees.

---

### **Code tables and data values and graphs**

When a column that is displayed in a graph has a code table, the graph displays the data values of the code table by default. To display the display values, call this function when you define the graph data.

---

**Examples** This expression returns the display value for the column `unit_measure`:

```
LookUpDisplay(unit_measure)
```

Assume the column `product_type` has a code table and you want to use it as a category for a graph. To display the product type descriptions instead of the data values in the categories, enter this expression in the Category option on the Data page in the Graph Object property sheet:

```
LookUpDisplay(product_type)
```

## Lower

Description Converts all the characters in a string to lowercase.

Syntax **Lower** ( *string* )

Argument	Description
<i>string</i>	The string you want to convert to lowercase letters

Return value String. Returns *string* with uppercase letters changed to lowercase if it succeeds and the empty string ("") if an error occurs.

Examples This expression returns babe ruth:

```
Lower ( "Babe Ruth" )
```

See also Upper

# Match

**Description** Determines whether a string's value contains a particular pattern of characters.

**Syntax** **Match** ( *string*, *textpattern* )

Argument	Description
<i>string</i>	The string in which you want to look for a pattern of characters
<i>textpattern</i>	A string whose value is the text pattern

**Return value** Boolean. Returns TRUE if *string* matches *textpattern* and FALSE if it does not. Match also returns FALSE if either argument has not been assigned a value or the pattern is invalid.

**Usage** Match enables you to evaluate whether a string contains a general pattern of characters. To find out whether a string contains a specific substring, use the Pos function.

*Textpattern* is similar to a regular expression. It consists of metacharacters, which have special meaning, and ordinary characters, which match themselves. You can specify that the string begin or end with one or more characters from a set, or that it contain any characters except those in a set.

A text pattern consists of **metacharacters**, which have special meaning in the match string, and **nonmetacharacters**, which match the characters themselves.

The following tables explain the meaning and use of these metacharacters:

Metacharacter	Meaning	Example
Caret (^)	Matches the beginning of a string	^C matches C at the beginning of a string
Dollar sign (\$)	Matches the end of a string	s\$ matches s at the end of a string
Period (.)	Matches any character	... matches three consecutive characters
Backslash (\)	Removes the following metacharacter's special characteristics so that it matches itself	\\$ matches \$

Metacharacter	Meaning	Example
Character class (a group of characters enclosed in square brackets [ ])	Matches any of the enclosed characters	[AEIOU] matches A, E, I, O, or U  You can use hyphens to abbreviate ranges of characters in a character class. For example, [A-Za-z] matches any letter
Complemented character class (first character inside the brackets is a caret)	Matches any character <i>not</i> in the group following the caret	[^0-9] matches any character except a digit, and [^A-Za-z] matches any character except a letter

The metacharacters asterisk (\*), plus (+), and question mark (?) are unary operators that are used to specify repetitions in a regular expression:

Metacharacter	Meaning	Example
* (asterisk)	Indicates zero or more occurrences	A* matches zero or more As (no As, A, AA, AAA, and so on)
+ (plus)	Indicates one or more occurrences	A+ matches one A or more than one A (A, AAA, and so on)
? (question mark)	Indicates zero or one occurrence	A? matches an empty string ("" ) or A

**Sample patterns** The following table shows various text patterns and sample text that matches each pattern:

This pattern	Matches
AB	Any string that contains AB; such as ABA, DEABC, graphAB_one
B*	Any string that contains 0 or more Bs; such as AC, B, BB, BBB, ABBBC, and so on
AB*C	Any string containing the pattern AC or ABC or ABBC, and so on (0 or more Bs)
AB+C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (1 or more Bs)
ABB*C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (1 B plus 0 or more Bs)

This pattern	Matches
<code>^AB</code>	Any string starting with AB
<code>AB?C</code>	Any string containing the pattern AC or ABC (0 or 1 B)
<code>^[ABC]</code>	Any string starting with A, B, or C
<code>[^ABC]</code>	A string containing any characters other than A, B, or C
<code>^[^abc]</code>	A string that begins with any character except a, b, or c
<code>^[^a-z]\$</code>	Any single-character string that is not a lowercase letter (^ and \$ indicate the beginning and end of the string)
<code>[A-Z]+</code>	Any string with one or more uppercase letters
<code>^[0-9]+\$</code>	Any string consisting only of digits
<code>^[0-9][0-9][0-9]\$</code>	Any string consisting of exactly three digits
<code>^([0-9][0-9][0-9])\$</code>	Any consisting of exactly three digits enclosed in parentheses

## Examples

This validation rule checks that the value the user entered begins with an uppercase letter. If value of the expression is false, the data fails validation:

```
Match(GetText(), "^[A-Z]")
```

## See also

Pos



# Max

**Description** Gets the maximum value in the specified column.

**Syntax** **Max** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the maximum value. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included when the maximum value is found. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The maximum value of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The maximum value of rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The maximum value of the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The maximum value of all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The maximum value in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The maximum value in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Max to consider only the distinct values in <i>column</i> when determining the largest value. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

**Return value** The data type of the column. Returns the maximum value in the rows of *column*. If you specify *range*, Max returns the maximum value in *column* in *range*.

## Usage

If you specify *range*, Max determines the maximum value in *column* in *range*. If you specify DISTINCT, Max returns the maximum distinct value in *column*, or if you specify *expressn*, the maximum distinct value in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

NULL values are ignored and are not considered in determining the maximum.

---

### **Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

## Examples

This expression returns the maximum of the values in the age column on the page:

```
Max(age for page)
```

This expression returns the maximum of the values in column 3 on the page:

```
Max(#3 for page)
```

This expression returns the maximum of the values in the column named age in group 1:

```
Max(age for group 1)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the maximum of the order amount for the distinct order numbers:

```
Max(order_amt for all DISTINCT order_nbr)
```

## See also

Min

## Median

### Description

Calculates the median of the values of the column. The median is the middle value in the set of values, for which there is an equal number of values greater and smaller than it.

### Syntax

**Median** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the median of the data values. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	The data that will be included in the median. For most presentation styles, values for <i>range</i> are: <ul style="list-style-type: none"> <li>◆ ALL — (Default) The median of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The median of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The median of the values in <i>column</i> on a page</li> </ul> For Crosstabs, specify CROSSTAB for <i>range</i> : <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The median of all values in <i>column</i> in the crosstab</li> </ul> For Graph and OLE objects, specify the type of object for <i>range</i> . The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are: <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The median of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The median of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Median to consider only the distinct values in <i>column</i> when determining the median. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

**Return value** The numeric data type of the column. Returns the median of the values of the rows in *range* if it succeeds and -1 if an error occurs.

**Usage** If you specify *range*, Median returns the median value of *column* in *range*. If you specify DISTINCT, Median returns the median value of the distinct values in *column*, or if you specify *expressn*, the median of *column* for each distinct value of *expressn*.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

In calculating the median, NULL values are ignored.

---

### Not in validation rules or filter expressions

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

**Examples** This expression returns the median of the values in the column named salary:

```
Median(salary)
```

This expression returns the median of the values in the column named salary of group 1:

```
Median(salary for group 1)
```

This expression returns the median of the values in column 5 on the current page:

```
Median(#5 for page)
```

This computed field returns Above Median if the median salary for the page is greater than the median for the report:

```
If(Median(salary for page) > Median(salary), "Above  
Median", " ")
```

This expression for a graph value sets the data value to the median value of the `sale_price` column:

```
Median(sale_price)
```

This expression for a graph value entered on the Data page in the Graph Object property sheet sets the data value to the median value of the `sale_price` column for the entire graph:

```
Median(sale_price for graph)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the median of the order amount for the distinct order numbers:

```
Median(order_amt for all DISTINCT order_nbr)
```

See also

Avg  
Mode

# Mid

## Description

Obtains a specified number of characters from a specified position in a string.

## Syntax

**Mid** ( *string*, *start* {, *length* } )

Argument	Description
<i>string</i>	The string from which you want characters returned
<i>start</i>	A long specifying the position of the first character you want returned (the position of the first character of the string is 1)
<i>length</i> (optional)	A long whose value is the number of characters you want returned. If you do not enter <i>length</i> or if <i>length</i> is greater than the number of characters to the right of <i>start</i> , Mid returns the remaining characters in the string

## Return value

String. Returns characters specified in *length* of *string* starting at character *start*. If *start* is greater than the number of characters in *string*, the Mid function returns the empty string (""). If *length* is greater than the number of characters remaining after the *start* character, Mid returns the remaining characters. The return string is not filled with spaces to make it the specified length.

## Examples

This expression returns "":

```
mid ("BABE RUTH", 40, 5)
```

This expression returns BE RUTH:

```
mid ("BABE RUTH", 3)
```

This expression in a computed field returns ACCESS DENIED if the fourth character in the column password is not R:

```
If (mid(password, 4, 1) = "R", "ENTER", "ACCESS  
DENIED")
```

To pass this validation rule the fourth character in the column password must be 6:

```
mid(password, 4, 1) = "6"
```

# Min

Description

Gets the minimum value in the specified column.

Syntax

**Min** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } }

Argument	Description
<i>column</i>	The column for which you want the minimum value. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	The data that will be included in the minimum. For most presentation styles, values for <i>range</i> are: <ul style="list-style-type: none"> <li>◆ ALL — (Default) The minimum of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The minimum of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The minimum of the values in <i>column</i> on a page</li> </ul> For Crosstabs, specify CROSSTAB for <i>range</i> : <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The minimum of all values in <i>column</i> in the crosstab</li> </ul> For Graph and OLE objects, specify the type of object for <i>range</i> . The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are: <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The minimum of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The minimum of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Min to consider only the distinct values in <i>column</i> when determining the minimum value. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Return value

The data type of the column. Returns the minimum value in the rows of *column*. If you specify *range*, Min returns the minimum value in the rows of *column* in *range*.

**Usage** If you specify *range*, Min determines the minimum value in *column* in *range*. If you specify DISTINCT, Min returns the minimum distinct value in *column*, or if you specify *expressn*, the minimum distinct value in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

NULL values are ignored and are not considered in determining the minimum.

---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

**Examples** This expression returns the minimum value in the column named age in group 2:

```
Min(age for group 2)
```

This expression returns the minimum of the values in column 3 on the page:

```
Min(#3 for page)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the minimum of the order amount for the distinct order numbers:

```
Min(order_amt for all DISTINCT order_nbr)
```

See also Max



# Minute

**Description** Obtains the number of minutes in the minutes portion of a time value.

**Syntax** **Minute** ( *time* )

<b>Argument</b>	<b>Description</b>
<i>time</i>	The time value from which you want the minutes

**Return value** Integer. Returns the minutes portion of *time* (00 to 59).

**Examples** This expression returns 1:

**Minute** (19:01:31)

**See also** Hour  
Second

# Mod

Description Obtains the remainder (modulus) of a division operation.

Syntax **Mod** ( *x*, *y* )

Argument	Description
<i>x</i>	The number you want to divide by <i>y</i>
<i>y</i>	The number you want to divide into <i>x</i>

Return value The data type of *x* or *y*, whichever data type is more precise.

Examples This expression returns 2:

**Mod** (20, 6)

This expression returns 1.5:

**Mod** (25.5, 4)

This expression returns 2.5:

**Mod** (25, 4.5)

# Mode

## Description

Calculates the mode of the values of the column. The mode is the most frequently occurring value.

## Syntax

**Mode** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } } )

Argument	Description
<i>column</i>	The column for which you want the mode of the data values. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	The data that will be included in the mode. For most presentation styles, values for <i>range</i> are: <ul style="list-style-type: none"> <li>◆ ALL — (Default) The mode of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The mode of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The mode of the values in <i>column</i> on a page</li> </ul> For Crosstabs, specify CROSSTAB for <i>range</i> : <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The mode of all values in <i>column</i> in the crosstab</li> </ul> For Graph and OLE objects, specify the type of object for <i>range</i> . The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are: <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The mode of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The mode of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Mode to consider only the distinct values in <i>column</i> when determining the mode. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

## Return value

The numeric data type of the column. Returns the mode of the values of the rows in *range* if it succeeds and -1 if an error occurs.

## Usage

If you specify *range*, Mode returns the mode of *column* in *range*. If you specify DISTINCT, Mode returns the mode of the distinct values in *column*, or if you specify *expressn*, the mode of *column* for each distinct value of *expressn*.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

In calculating the mode, NULL values are ignored.

---

**Not in validation rules or filter expressions**

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

## Examples

This expression returns the mode of the values in the column named salary:

```
Mode(salary)
```

This expression returns the mode of the values for group 1 in the column named salary:

```
Mode(salary for group 1)
```

This expression returns the mode of the values in column 5 on the current page:

```
Mode(#5 for page)
```

This computed field returns Above Mode if the mode of the salary for the page is greater than the mode for the report:

```
If(Mode(salary for page) > Mode(salary), "Above  
Mode", " ")
```

This expression for a graph value sets the data value to the mode of the sale\_price column:

```
Mode(sale_price)
```

This expression for a graph value entered on the Data page in the Graph Object property sheet sets the data value to the mode of the sale\_price column for the entire graph:

```
Mode(sale_price for graph)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the mode of the order amount for the distinct order numbers:

```
Mode(order_amt for all DISTINCT order_nbr)
```

See also

Avg  
Median

# Month

Description Gets the month of a date value.

Syntax **Month** ( *date* )

<b>Argument</b>	<b>Description</b>
<i>date</i>	The date from which you want the month

Return value Integer. Returns an integer (1 to 12) whose value is the month portion of *date*.

Examples This expression returns 1:

```
Month(1994-01-31)
```

This expression for a computed column returns Wrong Month if the month in the column expected\_grad\_date is not 6:

```
If(Month(expected_grad_date) = 6, "June", "Wrong  
Month")
```

This validation rule expression checks that the value of the month in the date in the column expected\_grad\_date is 6:

```
Month(expected_grad_date) = 6
```

See also  
Day  
Date  
Year

## Now

Description	Obtains the current time based on the system time of the client machine.
Syntax	<b>Now</b> ( )
Return value	Time. Returns the current time based on the system time of the client machine.
Usage	Use Now to compare a time to the system time or to display the system time on the screen. The timer interval specified for the DataWindow object, report, or form determines the frequency at which the value of Now is updated. For example, if the timer interval is 1 second, it is updated every second.
Examples	<p>This expression returns the current system time:</p> <pre>Now ( )</pre> <p>This expression sets the column value to 8:00 when the current system time is before 8:00 and to the current time if it is after 8:00:</p> <pre>If (Now() &lt; 08:00:00, '08:00:00', String(Now()))</pre>
See also	If Year

# Number

Description Converts a string to a number.

Syntax **Number** ( *string* )

Argument	Description
<i>string</i>	The string you want returned as a number

Return value A numeric data type. Returns the contents of *string* as a number. If *string* is not a valid number, Number returns 0.

Examples This expression converts the string 24 to a number:

**Number** ("24")

This expression for a computed field tests whether the value in the age column is greater than 55 and if so displays N/A; otherwise, it displays the value in age:

If (**Number**(age) > 55, "N/A", age)

This validation rule checks that the number the user entered is between 25,000 and 50,000:

**Number**(GetText())>25000 AND **Number** (GetText())<50000



## Page

Description	Gets the number of the current page.
Syntax	<b>Page</b> ( )
Return value	Integer. Returns the number of the current page.

---

### Calculating the page count

The vertical size of the paper less the top and bottom margins is used to calculate the page count. When the print orientation is landscape, the vertical size of the paper is the shorter dimension of the paper.

---

Examples This expression returns the number of the current page:

**Page** ( )

In the DataWindow object or report's footer band, this expression for a computed field displays a string showing the current page number and the total number of pages in the report. The result has the format *Page n of total*:

```
'Page ' + Page( ) + ' of ' + PageCount ( )
```

See also **PageAcross**  
**PageCount**  
**PageCountAcross**

## PageAcross

Description	Gets the number of the current horizontal page. For example, if a report is twice the width of the preview window and the window is scrolled horizontally to display the portion of the report that was outside the preview, PageAcross will return 2 because the current page is the second horizontal page.
Syntax	<b>PageAcross</b> ( )
Return value	Integer. Returns the number of the current horizontal page if it succeeds and -1 if an error occurs.
Examples	This expression returns the number of the current horizontal page:  <code>PageAcross ( )</code>
See also	Page PageCount PageCountAcross

## PageCount

Description	Gets the total number of pages when viewing a DataWindow object in Print Preview or when previewing a report. This number is also the number of printed pages if the DataWindow object or report is not wider than the preview window. If the DataWindow object or report is wider than the preview window, the number of printed pages will be greater than PageCount gets.
Syntax	<b>PageCount</b> ( )
Return value	Integer. Returns the total number of pages.
Usage	PageCount applies to Print Preview in the DataWindow painter and Preview in the Report painter. PageCount does not apply to Preview in the DataWindow painter.

---

### Calculating the page count

The vertical size of the paper less the top and bottom margins is used to calculate the page count. When the print orientation is landscape, the vertical size of the paper is the shorter dimension of the paper.

---

Examples	<p>This expression returns the number of pages:</p> <p style="text-align: center;"><b>PageCount</b> ( )</p> <p>In the DataWindow object or report's footer band, this expression for a computed field displays a string showing the current page number and the total number of pages in the report. The result has the format <i>Page n of total</i>:</p> <p style="text-align: center;">'Page ' + Page ( ) + ' of ' + <b>PageCount</b> ( )</p>
See also	<p>Page</p> <p>PageAcross</p> <p>PageCountAcross</p>

## PageCountAcross

Description	Gets the total number of horizontal pages when viewing a DataWindow object (in Print Preview) or report (in Preview) that is wider than the preview window.
Syntax	<b>PageCountAcross</b> ( )
Return value	Integer. Returns the total number of horizontal pages if it succeeds and -1 if an error occurs.
Usage	PageCountAcross applies to Print Preview in the DataWindow painter and Preview in the report painter. PageCountAcross does not apply to Preview in the DataWindow painter.
Examples	This expression returns the number of horizontal pages in the preview window:  <b>PageCountAcross</b> ( )
See also	Page PageAcross PageCount

# Percent

Description

Gets the percentage that the current value is of the total of the values in the column.

Syntax

**Percent** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the value of each row as a percentage of the total of the values of the column. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the percentage. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The percentage that the current value is of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The percentage that the current value is of rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The percentage that the current value is of the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The percentage that the current value is of all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The percentage that the current value is of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The percentage that the current value is of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Percent to consider only the distinct values in <i>column</i> when determining the percentage. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored

Argument	Description
<i>expressn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expressn</i> can be the name of a column, a function, or an expression

**Return value** A numeric data type (decimal, double, integer, long, or real). Returns the percentage the current row of *column* is of the total value of the column.

**Usage** Usually you use Percent in a column to display the percentage for each row. You can also use Percent in a header or trailer for a group. In the header, it displays the percentage for the first value in the group and in the trailer for the last value in the group.

If you specify *range*, Percent returns the percentage the current row of *column* relative to the total value of *range*. For example, if column 5 is salary, Percent(#5 for group 1) is equivalent to salary/(Sum(Salary for group 1)).

If you specify DISTINCT, Percent returns the percent the distinct value in *column* is of the total value of *column*, or if you specify *expressn*, the percent the value in *column* in the row in which the value of *expressn* is distinct is of the total for *column*.

---

#### Formatting the percent value

The percentage is displayed as a decimal value unless you specify a format for the result. A display format can be part of the computed field's definition.

---

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

NULL values are ignored and are not considered in the calculation.

**Not in validation rules, filter expressions, or crosstabs**

You cannot use Percent or other aggregate functions in validation rules or filter expressions. Percent does not work for crosstabs; specifying "for crosstab" as a range is not available for Percent.

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

**Examples**

This expression returns the value of each row in the column named salary as a percentage of the total of salary:

```
Percent(salary)
```

This expression returns the value of each row in the column named cost as a percentage of the total of cost in group 2:

```
Percent(cost for group 2)
```

This expression entered in the Value box on the Data tab page in the Graph Object property sheet returns the value of each row in the qty\_ordered as a percentage of the total for the column in the graph:

```
Percent(qty_ordered for graph)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the order amount as a percentage of the total order amount for the distinct order numbers:

```
Percent(order_amt for all DISTINCT order_nbr)
```

**See also**

CumulativePercent

# Pi

**Description** Multiplies pi by a specified number.

**Syntax** **Pi ( n )**

Argument	Description
<i>n</i>	The number you want to multiply by pi (3.14159265358979323...)

**Return value** Double. Returns the result of multiplying *n* by pi if it succeeds and -1 if an error occurs.

**Usage** Use Pi to convert angles to and from radians.

**Examples** This expression returns pi:

```
Pi (1)
```

Both these expressions return the area of a circle with the radius Rad:

```
Pi (1) * Rad^2
```

```
Pi (Rad^2)
```

This expression computes the cosine of a 45-degree angle:

```
Cos (45.0 * (Pi (2) / 360))
```

**See also**  
Cos  
Sin  
Tan



## Pos

Description	Finds one string within another string.								
Syntax	<p><b>Pos</b> ( <i>string1</i>, <i>string2</i>{, <i>start</i> } )</p> <table border="1"> <thead> <tr> <th>Argument</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>string1</i></td> <td>The string in which you want to find <i>string2</i></td> </tr> <tr> <td><i>string2</i></td> <td>The string you want to find in <i>string1</i></td> </tr> <tr> <td><i>start</i> (optional)</td> <td>A long indicating where the search will begin in <i>string</i>. The default is 1</td> </tr> </tbody> </table>	Argument	Description	<i>string1</i>	The string in which you want to find <i>string2</i>	<i>string2</i>	The string you want to find in <i>string1</i>	<i>start</i> (optional)	A long indicating where the search will begin in <i>string</i> . The default is 1
Argument	Description								
<i>string1</i>	The string in which you want to find <i>string2</i>								
<i>string2</i>	The string you want to find in <i>string1</i>								
<i>start</i> (optional)	A long indicating where the search will begin in <i>string</i> . The default is 1								
Return value	Long. Returns a long whose value is the starting position of the first occurrence of <i>string2</i> in <i>string1</i> after the position specified in <i>start</i> . If <i>string2</i> is not found in <i>string1</i> or if <i>start</i> is not within <i>string1</i> , Pos returns 0.								
Usage	The Pos function is case sensitive.								
Examples	<p>This expression returns the position of the letter <i>a</i> in the value of the last_name column:</p> <pre>Pos(last_name, "a")</pre> <p>This expression returns 6:</p> <pre>Pos("BABE RUTH", "RU")</pre> <p>This expression returns 1:</p> <pre>Pos("BABE RUTH", "B")</pre> <p>This expression returns 0 (because the case does not match):</p> <pre>Pos("BABE RUTH", "be")</pre> <p>This expression returns 0 (because it starts searching at position 5, after the occurrence of BE):</p> <pre>Pos("BABE RUTH", "BE", 5)</pre>								
See also	<p>Left Mid Right</p>								

# ProfileInt

Description

Obtains the integer value of a setting in the specified profile file.

Syntax

**ProfileInt** ( *filename*, *section*, *key*, *default* )

Argument	Description
<i>filename</i>	A string whose value is the name of the profile file. If you do not specify a full path, ProfileInt uses the operating system's standard file search order to find the file
<i>section</i>	A string whose value is the name of a group of related values in the profile file. In the file, section names are in square brackets. Do not include the brackets in <i>section</i> . <i>Section</i> is not case-sensitive
<i>key</i>	A string specifying the setting name in <i>section</i> whose value you want. The setting name is followed by an equal sign in the file. Do not include the equal sign in <i>key</i> . <i>Key</i> is not case-sensitive
<i>default</i>	An integer value that ProfileInt will return if <i>filename</i> is not found, if <i>section</i> or <i>key</i> does not exist in <i>filename</i> , or if the value of <i>key</i> cannot be converted to an integer

Return value

Integer. Returns *default* if *filename* is not found, *section* is not found in *filename*, *key* is not found in *section*, or the value of *key* is not an integer. Returns -1 if an error occurs.

Usage

Use ProfileInt and ProfileString to get configuration settings from a profile file you have designed for your application.

Examples

This example uses the following PROFILE.INI file:

```
[MyApp]
Maximized=1

[Security]
Class = 7
```

This expression tries to return the integer value of the keyword Minimized in section MyApp of file C:\PROFILE.INI. It returns 3 if there is no MyApp section or no Minimized keyword in the MyApp section. Based on the sample file above, it returns 3:

```
ProfileInt("C:\PROFILE.INI", "MyApp", "minimized", 3)
```

See also

`ProfileString`

# ProfileString

**Description** Obtains the string value of a setting in the specified profile file.

**Syntax** **ProfileString** ( *filename*, *section*, *key*, *default* )

Argument	Description
<i>filename</i>	A string whose value is the name of the profile file. If you do not specify a full path, ProfileString uses the operating system's standard file search order to find the file
<i>section</i>	A string whose value is the name of a group of related values in the profile file. In the file, section names are in square brackets. Do not include the brackets in <i>section</i> . <i>Section</i> is not case-sensitive
<i>key</i>	A string specifying the setting name in <i>section</i> whose value you want. The setting name is followed by an equal sign in the file. Do not include the equal sign in <i>key</i> . <i>Key</i> is not case-sensitive
<i>default</i>	A string value that ProfileString will return if <i>filename</i> is not found, if <i>section</i> or <i>key</i> does not exist in <i>filename</i> , or if the value of <i>key</i> cannot be converted to an integer

**Return value** String, with a maximum length of 4096 characters. Returns the string from *key* within *section* within *filename*. If *filename* is not found, *section* is not found in *filename*, or *key* is not found in *section*, ProfileString returns *default*. If an error occurs, it returns the empty string ("").

**Usage** Use ProfileInt and ProfileString to get configuration settings from a profile file you have designed for your application.

**Examples** This example uses the following section in PROFILE.INI file:

```
[Employee]
Name= "Smith"

[Dept]
Name= "Marketing"
```

This expression returns the string for the keyword Name in section Employee in file C:\PROFILE.INI. It returns None if the section or keyword does not exist. In this case it returns Smith:

```
ProfileString ("C:\PROFILE.INI", "Employee", "Name",
"None")
```

See also

ProfileInt

## Rand

**Description** Obtains a random whole number between 1 and a specified upper limit.

**Syntax** **Rand ( *n* )**

<b>Argument</b>	<b>Description</b>
<i>n</i>	The upper limit of the range of random numbers you want returned. The lower limit is always 1. The upper limit cannot exceed 32,767

**Return value** A numeric data type, the data type of *n*. Returns a random whole number between 1 and *n*.

**Usage** The sequence of numbers generated by repeated calls to the Rand function is a computer-generated pseudorandom sequence.

**Examples** This expression returns a random whole number between 1 and 10:

**Rand**(10)

## Real

Description Converts a string value to a real data type.

Syntax **Real** ( *string* )

Argument	Description
<i>string</i>	The string whose value you want to convert to a real

Return value Real. Returns the contents of a string as a real. If *string* is not a valid number, Real returns 0.

Examples This expression converts 24 to a real:

```
Real (" 24 ")
```

This expression returns the value in the column temp\_text as a real:

```
Real (temp_text)
```

## RelativeDate

**Description** Obtains the date that occurs a specified number of days after or before another date.

**Syntax** **RelativeDate** ( *date*, *n* )

<b>Argument</b>	<b>Description</b>
<i>date</i>	A date value
<i>n</i>	An integer indicating the number of days

**Return value** Date. Returns the date that occurs *n* days after *date* if *n* is greater than 0. Returns the date that occurs *n* days before *date* if *n* is less than 0.

**Examples** This expression returns 1990-02-10:

`RelativeDate(1990-01-31, 10)`

This expression returns 1990-01-21:

`RelativeDate(1990-01-31, -10)`

**See also** DaysAfter



## RelativeTime

**Description** Obtains a time that occurs a specified number of seconds after or before another time within a 24-hour period.

**Syntax** **RelativeTime** ( *time*, *n* )

Argument	Description
<i>time</i>	A time value
<i>n</i>	A long number of seconds

**Return value** Time. Returns the time that occurs *n* seconds after *time* if *n* is greater than 0. Returns the time that occurs *n* seconds before *time* if *n* is less than 0. The maximum return value is 23:59:59.

**Examples** This expression returns 19:01:41:

```
RelativeTime (19:01:31, 10)
```

This expression returns 19:01:21:

```
RelativeTime (19:01:31, -10)
```

**See also** SecondsAfter

# Replace

Description

Replaces a portion of one string with another.

Syntax

**Replace** ( *string1*, *start*, *n*, *string2* )

Argument	Description
<i>string1</i>	The string in which you want to replace characters with <i>string2</i>
<i>start</i>	A long whose value is the number of the first character you want replaced. (The first character in the string is number 1)
<i>n</i>	A long whose value is the number of characters you want to replace
<i>string2</i>	The string that will replace characters in <i>string1</i> . The number of characters in <i>string2</i> can be greater than, equal to, or fewer than the number of characters you are replacing

Return value

String. Returns the string with the characters replaced if it succeeds and the empty string ("") if it fails.

Usage

If the start position is beyond the end of the string, Replace appends *string2* to *string1*. If there are fewer characters after the start position than specified in *n*, Replace replaces all the characters to the right of character start.

If *n* is zero, then in effect Replace inserts *string2* into *string1*.

Examples

This expression changes the last two characters of the string David to e to make it Dave:

```
Replace("David", 4, 2, "e")
```

This expression returns BABY RUTH:

```
Replace("BABE RUTH", 1, 4, "BABY")
```

This expression returns Closed for the Winter:

```
Replace("Closed for Vacation", 12, 8, "the Winter")
```

## RGB

**Description** Calculates the long value that represents the color specified by numeric values for the red, green, and blue components of the color.

**Syntax** **RGB** ( *red*, *green*, *blue* )

Argument	Description
<i>red</i>	The integer value of the red component of the color
<i>green</i>	The integer value of the green component of the color
<i>blue</i>	The integer value of the blue component of the color

**Return value** Long. Returns the long that represents the color created by combining the values specified in red, green, and blue. If an error occurs, RGB returns NULL.

**Usage** The formula for combining the colors is:

$$\text{Red} + (256 * \text{Green}) + (65536 * \text{Blue})$$

Use RGB to obtain the long value required to set the color for text and drawing objects. You can also set an object's color to the long value that represents the color. The RGB function provides an easy way to calculate that value.

### Determining color components

The value of a component color is an integer between 0 and 255 that represents the amount of the component that is required to create the color you want. The lower the value, the darker the color; the higher the value, the lighter the color.

The following table lists red, green, and blue values for the 16 standard colors:

Color	Red value	Green value	Blue value
Black	0	0	0
White	255	255	255
Light Gray	192	192	192
Dark Gray	128	128	128
Red	255	0	0
Dark Red	128	0	0
Green	0	255	0

Color	Red value	Green value	Blue value
Dark Green	0	128	0
Blue	0	0	255
Dark Blue	0	0	128
Magenta	255	0	255
Dark Magenta	128	0	128
Cyan	0	255	255
Dark Cyan	0	128	128
Yellow	255	255	0
Brown	128	128	0

Examples

This expression returns as a long 8421376, which represents dark cyan:

```
RGB(0, 128, 128)
```

This expression for the Background.Color property of a salary column returns a long that represents red if an employee's salary is greater than \$50,000 and white if salary is less than or equal to \$50,000:

```
If(salary>50000, RGB(255, 0, 0), RGB(255, 255, 255))
```

## Right

**Description** Obtains a specified number of characters from the end of a string.

**Syntax** **Right** ( *string*, *n* )

Argument	Description
<i>string</i>	The string from which you want characters returned
<i>n</i>	A long whose value is the number of characters you want returned from the right end of <i>string</i>

**Return value** String. Returns the rightmost *n* characters in *string* if it succeeds and the empty string ("") if an error occurs.

If *n* is greater than or equal to the length of the string, Right returns the entire string. It does not add spaces to make the return value's length equal to *n*.

**Examples** This expression returns RUTH:

```
Right("BABE RUTH", 4)
```

This expression returns BABE RUTH:

```
Right("BABE RUTH", 75)
```

**See also**  
 Left  
 Mid  
 Pos

## RightTrim

**Description** Removes spaces from the end of a string.

**Syntax** **RightTrim** ( *string* )

<b>Argument</b>	<b>Description</b>
<i>string</i>	The string you want returned with trailing blanks deleted

**Return value** String. Returns a copy of *string* with trailing blanks deleted if it succeeds and the empty string ("") if an error occurs.

**Examples** This expression returns RUTH:

```
RightTrim("RUTH ")
```

**See also** LeftTrim  
Trim

# Round

**Description** Rounds a number to the specified number of decimal places.

**Syntax** **Round** ( *x* , *n* )

Argument	Description
<i>x</i>	The number you want to round
<i>n</i>	The number of decimal places to which you want to round <i>x</i>

**Return value** Decimal. If *n* is positive, Round returns *x* rounded to the specified number of decimal places. If *n* is negative, it returns *x* rounded to (- *n* +1) places before the decimal point. Returns -1 if it fails.

**Examples** This expression returns 9.62:

**Round** (9.624, 2)

This expression returns 9.63:

**Round** (9.625, 2)

This expression returns 9.600:

**Round** (9.6, 3)

This expression returns -9.63:

**Round** (-9.625, 2)

This expression returns -10:

**Round** (-9.625, -1)

**See also**

Ceiling  
Int  
Truncate

## RowCount

**Description** Obtains the number of rows that are currently available in the primary buffer.

**Syntax** **RowCount ( )**

**Return value** Long. Returns the number of rows that are currently available, 0 if no rows are currently available, and -1 if an error occurs.

**Examples** This expression in a computed field returns a warning if no data exists and the number of rows if there is data:

```
If(RowCount() = 0, "No Data", String(RowCount()))
```



## RowHeight

Description	Reports the height of a row associated with a band in a DataWindow object or a report.
Syntax	<b>RowHeight ( )</b>
Return value	Long. Returns the height of the row in the units specified for the DataWindow object or report if it succeeds, and -1 if an error occurs.
Usage	When you call RowHeight in a band other than the detail band, it reports on a row in the detail band. See GetRow for a table specifying which row is associated with each band for reporting purposes.
Examples	This expression for a computed field in the detail band displays the height of each row:  <code>RowHeight ( )</code>
See also	GetRow

## Second

Description Obtains the number of seconds in the seconds portion of a time value.

Syntax **Second** ( *time* )

Argument	Description
<i>time</i>	The time value from which you want the seconds

Return value Integer. Returns the seconds portion of *time* (00 to 59).

Examples This expression returns 31:

```
Second(19:01:31)
```

See also  
Hour  
Minute

## SecondsAfter

**Description** Gets the number of seconds one time occurs after another.

**Syntax** **SecondsAfter** ( *time1*, *time2* )

Argument	Description
<i>time1</i>	A time value that is the start time of the interval being measured
<i>time2</i>	A time value that is the end time of the interval

**Return value** Long. Returns the number of seconds *time2* occurs after *time1*. If *time2* occurs before *time1*, SecondsAfter returns a negative number.

**Examples** This expression returns 15:

**SecondsAfter** (21:15:30, 21:15:45)

This expression returns -15:

**SecondsAfter** (21:15:45, 21:15:30)

This expression returns 0:

**SecondsAfter** (21:15:45, 21:15:45)

**See also** DaysAfter

## Sign

**Description** Reports whether the number is negative, zero, or positive by checking its sign.

**Syntax** **Sign** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The number for which you want to determine the sign

**Return value** Integer. Returns a number (-1, 0, or 1) indicating the sign of *n*.

**Examples** This expression returns 1 (the number is positive):

**sign**(5)

This expression returns 0:

**sign**(0)

This expression returns -1 (the number is negative):

**sign**(-5)

## Sin

Description                      Calculates the sine of an angle.

Syntax                              **Sin** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The angle (in radians) for which you want the sine

Return value                      Double. Returns 1 if it succeeds and -1 if an error occurs.

Examples                            This expression returns .8414709848078965:

`sin(1)`

This expression returns 0:

`sin(0)`

This expression returns 0:

`sin(pi(1))`

See also                            Cos  
Pi  
Tan

## Small

### Description

Finds a small value at a specified ranking in a column (for example, third-smallest, fifth-smallest) and returns the value of another column or expression based on the result.

### Syntax

**Small** ( *returnexp*, *column*, *nbottom* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>returnexp</i>	The value you want returned when the small value is found. <i>Returnexp</i> includes a reference to a column, but not necessarily the column that is being evaluated for the small value, so that a value is returned from the same row that contains the small value
<i>column</i>	The column that contains the small value you are searching for. <i>Column</i> can be a column name or a column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
<i>nbottom</i>	The relationship of the small value to the column's smallest value. For example, when <i>nbottom</i> is 2, Small finds the second-smallest value
FOR <i>range</i> (optional)	<p>The data that will be included when finding the small value. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The small value of all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The small value of rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The small value of the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The small value of all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The small value in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The small value in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>

Argument	Description
DISTINCT (optional)	Causes Small to consider only the distinct values in <i>column</i> when determining the small value. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expressn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expressn</i> can be the name of a column, a function, or an expression

**Return value** The data type of *returnexp*. Returns the *nbottom*-smallest value if it succeeds and -1 if an error occurs.

**Usage** If you specify *range*, Small returns the value in *returnexp* when the value in *column* is the *nbottom*-smallest value in *range*. If you specify DISTINCT, Small returns *returnexp* when the value in *column* is the *nbottom*-smallest value of the distinct values in *column*, or if you specify *expressn*, the *nbottom*-smallest for each distinct value of *expressn*.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

---

**Min may be faster** If you don't need a return value from another column and you want to find the smallest value (*nbottom* = 1), use Min; it is faster.

**Not in validation rules or filter expressions** You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

**Examples** These expressions return the names of the salespersons with the three smallest sales (sum\_sales is the sum of the sales for each salesperson) in group 2, which might be the salesregion group. Note that sum\_sales contains the values being compared, but Small returns a value in the name column:

```
Small(name, sum_sales, 1 for group 2)  
Small(name, sum_sales, 2 for group 2)  
Small(name, sum_sales, 3 for group 2)
```

This example reports the salesperson with the third-smallest sales, considering only the first entry for each salesperson:

```
Small(name, sum_sales, 3 for all DISTINCT sum_sales)
```

See also

Large



# Space

**Description** Builds a string of the specified length whose value consists of spaces.

**Syntax**

**Space** ( *n* )

Argument	Description
<i>n</i>	A long whose value is the length of the string you want filled with spaces

**Return value**

String. Returns a string filled with *n* spaces if it succeeds and the empty string ("") if an error occurs.

**Examples**

This expression for a computed field returns 10 spaces in the computed field if the value of the rating column is Top Secret; otherwise, it returns the value in rating:

```
If(rating = "Top Secret", Space(10), rating)
```

**See also**

Fill

# Sqrt

Description                      Calculates the square root of a number.

Syntax                              **Sqrt** ( *n* )

<b>Argument</b>	<b>Description</b>
<i>n</i>	The number for which you want the square root

Return value                      Double. Returns the square root of *n*.

Usage                                Sqrt(*n*) is the same as  $n^{.5}$ .

Taking the square root of a negative number causes an execution error.

Examples                            This expression returns 1.414213562373095:

**Sqrt** (2)

This expression results in an error at execution time:

**Sqrt** (-2)

## StDev

### Description

Calculates an estimate of the standard deviation for the specified column. Standard deviation is a measurement of how widely values vary from average.

### Syntax

**StDev** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want an estimate for the standard deviation of the values in the rows. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the estimate of the standard deviation. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The estimate of the standard deviation for all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The estimate of the standard deviation for values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The estimate of the standard deviation for the values in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The estimate of the standard deviation for all values in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The estimate of the standard deviation for values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The estimate of the standard deviation for values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes StDev to consider only the distinct values in <i>column</i> when determining the standard deviation. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Return value

Double. Returns an estimate of the standard deviation for *column*.

Usage

If you specify *range*, StDev returns an estimate for the standard deviation of *column* within *range*. If you specify DISTINCT, StDev returns an estimate of the standard deviation for the distinct values in *column* or if you specify *expressn*, the estimate of the standard deviation of the rows in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data tab page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

---

**Estimating or calculating actual standard deviation** StDev assumes that the values in *column* are a sample of the values in the rows in the column in the database table. If you selected all the rows in the column in the DataWindow object's SELECT statement, use StDevP to compute the standard deviation of the population.

**Not in validation rules or filter expressions** You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

Examples

These examples all assume that the SELECT statement did not retrieve all the rows in the database table. StDev is intended to work with a subset of rows, which is a sample of the full set of data.

This expression returns an estimate for standard deviation of the values in the column named salary:

```
StDev(salary)
```

This expression returns an estimate for standard deviation of the values in the column named salary in group 1:

```
StDev(salary for group 1)
```

This expression returns an estimate for standard deviation of the values in column 4 on the page:

```
StDev(#4 for page)
```

This expression entered in the Value box on the Data tab page in the Graph Object property sheet returns an estimate for standard deviation of the values in qty\_used column in the graph:

```
StDev(qty_used for graph)
```

This expression for a computed field in a crosstab returns the estimate for standard deviation of the values in qty\_ordered column in the crosstab:

```
StDev(qty_ordered for crosstab)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the estimated standard deviation of the order amount for the distinct order numbers:

```
StDev(order_amt for all DISTINCT order_nbr)
```

See also

StDevP  
Var

# StDevP

## Description

Calculates the standard deviation for the specified column. Standard deviation is a measurement of how widely values vary from average.

## Syntax

**StDevP** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the standard deviation of the values in the rows. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the standard deviation. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The standard deviation for all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The standard deviation for values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The standard deviation for the values in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The standard deviation for all values in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The standard deviation for values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The standard deviation for values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes StDevP to consider only the distinct values in <i>column</i> when determining the standard deviation. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Return value	Double. Returns the standard deviation for <i>column</i> .
Usage	<p>If you specify <i>range</i>, StDevP returns the standard deviation for <i>column</i> within <i>range</i>. If you specify DISTINCT, StDev returns an estimate of the standard deviation for the distinct values in <i>column</i> or if you specify <i>expressn</i>, the estimate of the standard deviation of the rows in <i>column</i> where the value of <i>expressn</i> is distinct.</p> <p>For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data tab page (the Range property), and the aggregation function uses that range. Settings for Rows include:</p> <ul style="list-style-type: none"> <li>◆ For the Graph or OLE presentation style, Rows is always All.</li> <li>◆ For Graph objects, Rows can be All, Page, or Group.</li> <li>◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.</li> </ul>

---

**Estimating or calculating actual standard deviation** StDevP assumes that the values in *column* are the values in all the rows in the column in the database table. If you did not select all rows in the column in the SELECT statement, use StDev to compute an estimate of the standard deviation of a sample.

**Not in validation rules or filter expressions** You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

#### Examples

These examples all assume that the SELECT statement retrieved all rows in the database table. StDevP is intended to work with a full set of data, not a subset.

This expression returns the standard deviation of the values in the column named salary:

```
StDevP (salary)
```

This expression returns the standard deviation of the values in group 1 in the column named salary:

```
StDevP (salary for group 1)
```

This expression returns the standard deviation of the values in column 4 on the page:

**StDevP** (#4 for page)

This expression entered in the Value box on the Data tab page in the Graph Object property sheet returns the standard deviation of the values in qty\_ordered column in the graph:

**StDevP**(qty\_ordered for graph)

This expression for a computed field in a crosstab returns the standard deviation of the values in qty\_ordered column in the crosstab:

**StDevP**(qty\_ordered for crosstab)

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the standard deviation of the order amount for the distinct order numbers:

**StDevP**(order\_amt for all DISTINCT order\_nbr)

See also

StDev

VarP



# String

## Description

Formats data as a string according to a specified display format mask. You can convert and format date, DateTime, numeric, and time data. You can also apply a display format to a string.

## Syntax

**String** ( *data* {, *format* } )

Argument	Description
<i>data</i>	The data you want returned as a string with the specified formatting. <i>Data</i> can have a date, DateTime, numeric, time, or string data type
<i>format</i> (optional)	A string of the display masks you want to use to format the data. The masks consist of formatting information specific to the data type of <i>data</i> . If <i>data</i> is type string, <i>format</i> is required  The format string can consist of more than one mask, depending on the data type of <i>data</i> . Each mask is separated by a semicolon. See Usage for details on each data type

## Return value

String. Returns *data* in the specified format if it succeeds and the empty string ("") if the data type of *data* does not match the type of display mask specified or *format* is not a valid mask.

## Usage

For date, DateTime, numeric, and time data, the system's default format is used for the returned string if you don't specify a format. For numeric data, the default format is the [General] format.

For string data, a display format mask is required. (Otherwise, the function would have nothing to do.)

The format can consist of one or more masks:

- ◆ Formats for date, DateTime, string, and time data can include one or two masks. The first mask is the format for the data; the second mask is the format for a null value.
- ◆ Formats for numeric data can have up to four masks. A format with a single mask handles both positive and negative data. If there are additional masks, the first mask is for positive values, and the additional masks are for negative, zero, and NULL values.

A format can include color specifications, which are displayed in the DataWindow object, report, or form.

If the display format doesn't match the data type, PowerBuilder and InfoMaker will try to apply the mask, which can produce unpredictable results.

FOR INFO For information about localized deployment kits, which are available in French, German, Italian, Spanish, Dutch, Danish, Norwegian, and Swedish, see "The deployed application" on page 649.

Examples

This expression returns Jan 31, 1998:

```
String(1998-01-31, "mmm dd, yyyy")
```

This expression returns Jan 31, 1998, 6 hrs and 8 min:

```
String(1998-01-31 06:08:00, 'mmm dd, yyyy, h "hrs  
and" m "min"')
```

This expression:

```
String(nbr, "0000;(000);****;empty")
```

returns:

```
0123 if nbr is 123  
(123) if nbr is -123  
**** if nbr is 0  
empty if nbr is NULL
```

This expression returns A-B-C:

```
String("ABC", "@-@-@")
```

This expression returns A\*B:

```
String("ABC", "@*@" )
```

This expression returns ABC:

```
String("ABC", "@@@")
```

This expression returns a space:

```
String("ABC", " ")
```

This expression returns 6 hrs and 8 min:

```
String(06:08:02, 'h "hrs and" m "min"')
```

This expression returns 08:06:04 pm:

```
String(20:06:04, "hh:mm:ss am/pm")
```

This expression returns 8:06:04 am:

```
String(08:06:04, "h:mm:ss am/pm")
```

This expression returns 6:11:25.300000:

```
String(6:11:25.300000, "h:mm:ss.ffffff")
```

## Sum

Description

Calculates the sum of the values in the specified column.

Syntax

**Sum** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which you want the sum of the data values. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	The data that will be included in the sum. For most presentation styles, values for <i>range</i> are: <ul style="list-style-type: none"> <li>◆ ALL — (Default) The sum of all values in <i>column</i></li> <li>◆ GROUP <i>n</i> — The sum of values in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The sum of the values in <i>column</i> on a page</li> </ul> For Crosstabs, specify CROSSTAB for <i>range</i> : <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The sum of all values in <i>column</i> in the crosstab</li> </ul> For Graph and OLE objects, specify the type of object for <i>range</i> . The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are: <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The sum of values in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The sum of values in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Sum to consider only the distinct values in <i>column</i> when determining the sum. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

Return value

The appropriate numeric data type. Returns the sum of the data values in *column*.

### Usage

If you specify *range*, Sum returns the sum of the values in *column* within *range*. If you specify DISTINCT, Sum returns the sum of the distinct values in *column*, or if you specify *expressn*, the sum of the values of *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

NULL values are ignored and are not included in the calculation.

---

### Not in validation rules or filter expressions

You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

### Examples

This expression returns the sum of the values in group 1 in the column named salary:

```
Sum(salary for group 1)
```

This expression returns the sum of the values in column 4 on the page:

```
Sum(#4 for page)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the sum of the order amount for the distinct order numbers:

```
Sum(order_amt for all DISTINCT order_nbr)
```

# Tan

Description Calculates the tangent of an angle.

Syntax **Tan ( *n* )**

<b>Argument</b>	<b>Description</b>
<i>n</i>	The angle (in radians) for which you want the tangent

Return value Double. Returns the tangent of *n* if it succeeds and -1 if an error occurs.

Examples Both these expressions return 0:

```
Tan(0)  
Tan(Pi(1))
```

This expression returns 1.55741:

```
Tan(1)
```

See also  
Cos  
Pi  
Sin

## Time

**Description** Converts a string to a time data type.

**Syntax** **Time** ( *string* )

Argument	Description
<i>string</i>	A string containing a valid time (such as 8 AM or 10:25) that you want returned as a time data type. Only the hour is required; you do not have to include the minutes, seconds, or microseconds of the time or AM or PM. The default value for minutes and seconds is 00 and for microseconds is 000000. AM or PM is determined automatically

**Return value** Time. Returns the time in *string* as a time data type. If *string* does not contain a valid time, Time returns 00:00:00.

**Examples** This expression returns the time data type for 45 seconds before midnight (23:59:15):

```
Time ("23:59:15")
```

This expression for a computed field returns the value in the `time_received` column as a value of type time if `time_received` is not the empty string. Otherwise, it returns 00:00:00:

```
If (time_received = "" , 00:00:00, Time(time_received))
```

This example is similar to the previous one, except that it returns 00:00:00 if `time_received` contains a NULL value:

```
If (IsNull(time_received), 00:00:00, Time(time_received))
```

## Today

Description	Obtains the system date and time.
Syntax	<b>Today ( )</b>
Return value	DateTime. Returns the current system date and time.
Usage	To display both the date and the time, a computed field must have a display format that includes the time.
Examples	This expression for a computed field displays the date and time when the display format for the field is "mm/dd/yy hh:mm":  <b>Today ( )</b>
See also	Now

# Trim

Description Removes leading and trailing spaces from a string.

Syntax **Trim** ( *string* )

Argument	Description
<i>string</i>	The string you want returned with leading and trailing spaces deleted

Return value String. Returns a copy of *string* with all leading and trailing spaces deleted if it succeeds and the empty string ("") if an error occurs.

Usage Trim is useful for removing spaces that a user may have typed before or after newly entered data.

Examples This expression returns BABE RUTH:

```
Trim( " BABE RUTH " )
```

See also LeftTrim  
RightTrim



# Truncate

**Description** Truncates a number to the specified number of decimal places.

**Syntax** **Truncate** ( *x*, *n* )

Argument	Description
<i>x</i>	The number you want to truncate
<i>n</i>	The number of decimal places to which you want to truncate <i>x</i>

**Return value** The data type of *x*. If *n* is positive, returns *x* truncated to the specified number of decimal places. If *n* is negative, returns *x* truncated to ( $-n + 1$ ) places before the decimal point. Returns -1 if it fails.

**Examples** This expression returns 9.2:

**Truncate** (9.22, 1)

This expression returns 9.2:

**Truncate** (9.28, 1)

This expression returns 9:

**Truncate** (9.9, 0)

This expression returns -9.2:

**Truncate** (-9.29, 1)

This expression returns 0:

**Truncate** (9.2, -1)

This expression returns 50:

**Truncate** (54, -1)

**See also**

Ceiling  
Int  
Round

# Upper

Description Converts all characters in a string to uppercase letters.

Syntax **Upper** ( *string* )

Argument	Description
<i>string</i>	The string you want to convert to uppercase letters

Return value String. Returns *string* with lowercase letters changed to uppercase if it succeeds and the empty string ("") if an error occurs.

Examples This expression returns BABE RUTH:

```
Upper ("Babe Ruth")
```

See also Lower

# Var

## Description

Calculates an estimate of the variance for the specified column. The variance is the square of the standard deviation.

## Syntax

**Var** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } } )

Argument	Description
<i>column</i>	The column for which for which you want an estimate for the variance of the values in the rows. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the estimate of the variance. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The estimate of the variance for all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The estimate of the variance for rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The estimate of the variance for the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The estimate of the variance for all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The estimate of the variance for rows in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The estimate of the variance for rows in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes Var to consider only the distinct values in <i>column</i> when determining the variance. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

**Return value** Double. Returns an estimate for the variance for *column*. If you specify *group*, Var returns an estimate for the variance for *column* within *group*.

**Usage** If you specify *range*, Var returns an estimate for the variance for *column* within *range*. If you specify DISTINCT, Var returns the variance for the distinct values in *column* or if you specify *expressn*, the estimate for the variance of the rows in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

---

**Estimating variance or calculating actual variance** Var assumes that the values in *column* are a sample of the values in rows in the column in the database table. If you select all rows in the column in the SELECT statement, use VarP to compute the variance of a population.

**Not in validation rules or filter expressions** You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

**Examples** These examples all assume that the SELECT statement did not retrieve all of the rows in the database table. Var is intended to work with a subset of rows, which is a sample of the full set of data.

This expression returns an estimate for the variance of the values in the column named salary:

```
Var(salary)
```

This expression returns an estimate for the variance of the values in the column named salary in group 1:

```
Var(salary for group 1)
```

This expression entered in the Value box on the Data property page in the Graph Object property sheet returns an estimate for the variance of the values in quantity column in the graph:

```
Var(quantity for graph)
```

This expression for a computed field in a crosstab returns an estimate for the variance of the values in the quantity column in the crosstab:

```
Var(quantity for crosstab)
```

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the estimate for the variance of the order amount for the distinct order numbers:

```
Var(order_amt for all DISTINCT order_nbr)
```

See also

StDev  
VarP

# VarP

## Description

Calculates the variance for the specified column. The variance is the square of the standard deviation.

## Syntax

**VarP** ( *column* { FOR *range* { DISTINCT { *expres1* {, *expres2* {, ... } } } } } )

Argument	Description
<i>column</i>	The column for which for which you want the variance of the values in the rows. <i>Column</i> can be the column name or the column number preceded by a pound sign (#). <i>Column</i> can also be an expression that includes a reference to the column. The data type of <i>column</i> must be numeric
FOR <i>range</i> (optional)	<p>The data that will be included in the variance. For most presentation styles, values for <i>range</i> are:</p> <ul style="list-style-type: none"> <li>◆ ALL — (Default) The variance for all rows in <i>column</i></li> <li>◆ GROUP <i>n</i> — The variance for rows in <i>column</i> in the specified group. Specify the keyword GROUP followed by the group number: for example, GROUP 1</li> <li>◆ PAGE — The variance for the rows in <i>column</i> on a page</li> </ul> <p>For Crosstabs, specify CROSSTAB for <i>range</i>:</p> <ul style="list-style-type: none"> <li>◆ CROSSTAB — (Crosstabs only) The variance for all rows in <i>column</i> in the crosstab</li> </ul> <p>For Graph and OLE objects, specify the type of object for <i>range</i>. The values to be aggregated are determined by the range specified in the object definition. (See Usage for more information.) Values are:</p> <ul style="list-style-type: none"> <li>◆ GRAPH — (Graphs only) The variance for rows in <i>column</i> in the range specified for the Rows option of the graph</li> <li>◆ OBJECT — (OLE objects only) The variance for rows in <i>column</i> in the range specified for the Rows option of the OLE object</li> </ul>
DISTINCT (optional)	Causes VarP to consider only the distinct values in <i>column</i> when determining the variance. For a value of <i>column</i> , the first row found with the value is used and other rows that have the same value are ignored
<i>expresn</i> (optional)	One or more expressions that you want to evaluate to determine distinct rows. <i>Expresn</i> can be the name of a column, a function, or an expression

## Return value

Double. Returns the variance for *column*. If you specify *group*, Var returns the variance for *column* within *range*.

## Usage

If you specify *range*, VarP returns the variance for *column* within *range*. If you specify DISTINCT, VarP returns the variance for the distinct values in *column* or if you specify *expressn*, the variance of the rows in *column* where the value of *expressn* is distinct.

For graphs and OLE objects, you do not select the range when you call the function. The range for the object has already been determined by the Rows setting on the Data property page (the Range property), and the aggregation function uses that range. Settings for Rows include:

- ◆ For the Graph or OLE presentation style, Rows is always All.
- ◆ For Graph objects, Rows can be All, Page, or Group.
- ◆ For OLE objects, Rows can be All, Current Row, Page, or Group. The available choices depend on the layer the object occupies.

---

**Estimating variance or calculating actual variance** VarP assumes that the values in *column* are the values in all rows in the column in the database table. If you did not select all the rows in the column in the SELECT statement, use Var to compute an estimate of the variance of a sample.

**Not in validation rules or filter expressions** You cannot use this or other aggregate functions in validation rules or filter expressions.

---

Using an aggregate function cancels the effect of setting Retrieve Rows As Needed in the DataWindow painter and Report painter. To do the aggregation, a DataWindow object or report always retrieves all rows.

## Examples

These examples all assume that the SELECT statement retrieved all rows in the database table. VarP is intended to work with a full set of data, not a subset.

This expression returns the variance of the values in the column named salary:

```
VarP(salary)
```

This expression returns the variance of the values in group 1 in the column named salary:

```
VarP(salary for group 1)
```

This expression returns the variance of the values in column 4 on the page:

```
VarP(#4 for page)
```

This expression entered in the Value box on the Data property page in the Graph Object property sheet returns the variance of the values in quantity column in the graph:

**VarP**(quantity for graph)

This expression for a computed field in a crosstab returns the variance of the values in quantity column in the crosstab:

**VarP**(quantity for crosstab)

Assuming a DataWindow object, report, or form displays the order number, amount, and line items for each order, this computed field returns the variance of the order amount for the distinct order numbers:

**VarP**(order\_amt for all DISTINCT order\_nbr)

See also

StDevP

Var



## WordCap

**Description** Sets the first letter of each word in a string to a capital letter and all other letters to lowercase (for example, ROBERT E. LEE would be Robert E. Lee).

**Syntax** **WordCap** ( *string* )

Argument	Description
<i>string</i>	A string or expression that evaluates to a string that you want to display with initial capital letters (for example, Monday Morning)

**Return value** String. Returns *string* with the first letter of each word set to uppercase and the remaining letters lowercase if it succeeds and NULL if an error occurs.

**Examples** This expression returns Boston, Massachusetts:

```
WordCap ("boston, MASSACHUSETTS")
```

This expression concatenates the characters in the emp\_fname and emp\_lname columns and makes the first letter of each word uppercase:

```
WordCap (emp_fname + " " + emp_lname)
```

## Year

Description Gets the year of a date value.

Syntax **Year** ( *date* )

<b>Argument</b>	<b>Description</b>
<i>date</i>	The date value from which you want the year

Return value Integer. Returns an integer whose value is a 4-digit year adapted from the year portion of *date* if it succeeds and 1900 if an error occurs.

If the year is two digits, then PowerBuilder and InfoMaker choose the century as follows. If the year is between 00 to 49, PowerBuilder and InfoMaker assume 20 as the first two digits; if it is between 50 and 99, PowerBuilder and InfoMaker assume 19.

Usage Obtains the year portion of *date*. PowerBuilder and InfoMaker handle years from 1000 to 3000 inclusive.

If your data includes dates before 1950, such as birth dates, always specify a 4-digit year so that Year (and other functions, such as Sort) interpret the date as intended.

Examples This expression returns 1995:

```
Year (1995-01-31)
```

See also Day  
Month

# A P P E N D I X



# The Powersoft Repository

About this appendix

This appendix describes each column in each table in the Powersoft repository.

Contents

<b>Topic</b>	<b>Page</b>
About the Powersoft repository	866
The PBCatTbl table	867
The PBCatCol table	869
The PBCatFmt table	870
The PBCatVld table	871
The PBCatEdt table	872

## About the Powersoft repository

InfoMaker stores information you provide for a database table (such as the text to use for labels and headings for the columns, validation rules, display formats, and edit styles) in system tables in your database. These system tables are called the Powersoft **repository**. The repository tables contain all the information related to the extended attributes for the tables and columns in the database. These extended attributes are used in reports.

### The system tables

There are five Powersoft system tables:

Table	Contains information about
PBCatTbl	Tables in the database
PBCatCol	Columns in the database
PBCatFmt	Display formats
PBCatVld	Validation rules
PBCatEdt	Edit styles

### What to do with the tables

You can open and look at these tables in the Database painter just like other tables. You might want to create a report of the repository information used in your database by building a report whose data source is the repository tables.

---

#### Caution

You should not change the values in the repository tables. InfoMaker maintains this information automatically whenever you change information for a table or column in the Database painter.

---

## The PBCatTbl table

Column	Column name	Description
1	pbt_tnam	Table name
2	pbt_tid	SQL Server Object ID of table (used for SQL Server only)
3	pbt_ownr	Table owner
4	pbd_fhgt	Data font height, PowerBuilder units
5	pbd_fwgt	Data font stroke weight (400=Normal, 700=Bold)
6	pbd_fitl	Data font Italic (Y=Yes, N=No)
7	pbd_funl	Data font Underline (Y=Yes, N=No)
8	pbd_fchr	Data font character set (0=ANSI, 2=Symbol, 255=OEM)
9	pbd_fptc	Data font pitch and family (see note)
10	pbd_ffce	Data font typeface
11	pbh_fhgt	Headings font height, PowerBuilder units
12	pbh_fwgt	Headings font stroke weight (400=Normal, 700=Bold)
13	pbh_fitl	Headings font Italic (Y=Yes, N=No)
14	pbh_funl	Headings font Underline (Y=Yes, N=No)
15	pbh_fchr	Headings font character set (0=ANSI, 2=Symbol, 255=OEM)
16	pbh_fptc	Headings font pitch and family (see note)
17	pbh_ffce	Headings font typeface
18	pbl_fhgt	Labels font height, PowerBuilder units
19	pbl_fwgt	Labels font stroke weight (400=Normal, 700=Bold)
20	pbl_fitl	Labels font Italic (Y=Yes, N=No)
21	pbl_funl	Labels font Underline (Y=Yes, N=No)
22	pbl_fchr	Labels font character set (0=ANSI, 2=Symbol, 255=OEM)

<b>Column</b>	<b>Column name</b>	<b>Description</b>
23	pbl_fptc	Labels font pitch and family (see note)
24	pbl_ffce	Labels font typeface
25	pbt_cmnt	Table comments

---

**About font pitch and family**

Font pitch and family is a number obtained by adding together two constants:

*Pitch*: 0=Default, 1=Fixed, 2=Variable

*Family*: 0=Don't Care, 16=Roman, 32=Swiss, 48=Modern, 64=Script, 80=Decorative

---



## The PBCatCol table

Column	Column name	Description
1	pbc_tnam	Table name
2	pbc_tid	SQL Server Object ID of table (used for SQL Server only)
3	pbc_ownr	Table owner
4	pbc_cnam	Column name
5	pbc_cid	SQL Server Column ID (used for SQL Server only)
6	pbc_labl	Label
7	pbc_lpos	Label position (23=Left, 24=Right)
8	pbc_hdr	Heading
9	pbc_hpos	Heading position (23=Left, 24=Right, 25=Center)
10	pbc_jtfy	Justification (23=Left, 24=Right)
11	pbc_mask	Display format name
12	pbc_case	Case (26=Actual, 27=UPPER, 28=lower)
13	pbc_hght	Column height, PowerBuilder units
14	pbc_wdth	Column width, PowerBuilder units
15	pbc_ptrn	Validation rule name
16	pbc_bmap	Bitmap/picture (Y=Yes, N=No)
17	pbc_init	Initial value
18	pbc_cmnt	Column comments
19	pbc_edit	Edit style name
20	pbc_tag	(Reserved)

## The PBCatFmt table

Column	Column name	Description
1	pbf_name	Display format name
2	pbf_frmt	Display format
3	pbf_type	Data type to which format applies
4	pbf_cntr	Concurrent-usage flag

## The PBCatVId table

Column	Column name	Description
1	pbv_name	Validation rule name
2	pbv_vald	Validation rule
3	pbv_type	Data type to which validation rule applies
4	pbv_cntr	Concurrent-usage flag
5	pbv_msg	Validation error message

## The PBCatEdt table

Column	Column name	Description
1	pbe_name	Edit style name
2	pbe_edit	Format string (edit style type dependent; see below)
3	pbe_type	Edit style type (see below)
4	pbe_cntr	Revision counter (increments each time edit style is altered)
5	pbe_seqn	Row sequence number for edit types requiring more than one row in PBCatEdt table
6	pbe_flag	Edit style flag (edit style type dependent; see below)
7	pbe_work	Extra field (edit style type dependent; see below)

## Available edit style types

The following edit style types are available:

Edit style type	pbe_type value (column 3)
CheckBox	85
RadioButton	86
DropDownListBox	87
DropDownDataWindow	88
Edit	89
Edit Mask	90

## CheckBox edit style (code 85)

Here is a sample row in the PBCatEdt table for a CheckBox edit style:

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>Text</i>	85	1	1	<i>Flag</i>	
MyEdit	<i>OnValue</i>	85	1	2	0	

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>OffValue</i>	85	1	3	0	
MyEdit	<i>ThirdValue</i>	85	1	4	0	

where:

Value	Meaning
<i>Text</i>	CheckBox text
<i>OnValue</i>	Data value for On state
<i>OffValue</i>	Data value for Off state
<i>ThirdValue</i>	Data value for Third state (this row exists only if 3 State is checked for the edit style—bit 30 of <i>Flag</i> is 1)
<i>Flag</i>	<p>32-bit flag. Low-order four hex digits are generic edit type; high-order four are styles within the type. A 1 in any bit indicates the corresponding style is checked. A 0 in any bit indicates the corresponding style is unchecked</p> <p>Bit 31: Left Text            Bit 30: 3 State            Bit 29: 3D            Bit 28: Scale Box            Bits 27 – 16 (3 hex digits): Not used (set to 0)            Bits 15 – 4 (3 hex digits): Always 0 for CheckBox edit style            Bit 3: Always 0 for CheckBox edit style            Bit 2: Always 1 for CheckBox edit style            Bit 1: Always 0 for CheckBox edit style            Bit 0: Always 0 for CheckBox edit style</p>

## RadioButton edit style (code 86)

Here is a sample row in the PBCatEdt table for a RadioButton edit style:

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>Columns</i>	86	1	1	<i>Flag</i>	
MyEdit	<i>Display1</i>	86	1	2	0	
MyEdit	<i>Data1</i>	86	1	3	0	
MyEdit	<i>Display2</i>	86	1	4	0	
MyEdit	<i>Data2</i>	86	1	5	0	

where:

Value	Meaning
<i>Columns</i>	Character representation (in decimal) of number of columns (buttons) across
<i>Display1</i>	Display value for first button
<i>Data1</i>	Data value for first button
<i>Display2</i>	Display value for second button
<i>Data2</i>	Data value for second button
	<p><b>About the display and data values</b></p> <p>Display and data values are repeated in pairs for each radio button defined in the edit style</p>
<i>Flag</i>	<p>32-bit flag. Low-order four hex digits are generic edit type; high-order four are styles within the type. A 1 in any bit indicates the corresponding style is checked. A 0 in any bit indicates the corresponding style is unchecked</p> <p>Bit 31: Left Text            Bit 30: 3D            Bit 29: Scale Circles            Bit 38: Not used (set to 0)            Bits 27 – 16 (3 hex digits): Not used (set to 0)            Bits 15 – 4 (3 hex digits): Always 0 for RadioButton edit style            Bit 3: Always 1 for RadioButton edit style            Bit 2: Always 0 for RadioButton edit style            Bit 1: Always 0 for RadioButton edit style            Bit 0: Always 0 for RadioButton edit style</p>

### DropDownListBox edit style (code 87)

Here is a sample row in the PBCatEdt table for a DropDownListBox edit style:

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>Limit</i>	87	1	1	<i>Flag</i>	<i>Key</i>
MyEdit	<i>Display1</i>	87	1	2	0	
MyEdit	<i>Data1</i>	87	1	3	0	
MyEdit	<i>Display2</i>	87	1	4	0	
MyEdit	<i>Data2</i>	87	1	5	0	

where:

<b>Value</b>	<b>Meaning</b>
<i>Limit</i>	Character representation (in decimal) of the <i>Limit</i> value
<i>Key</i>	One-character accelerator key
<i>Display1</i>	Display value for first entry in code table
<i>Data1</i>	Data value for first entry in code table
<i>Display2</i>	Display value for second entry in code table
<i>Data2</i>	Data value for second entry in code table
	<p><b>About the display and data values</b>            Display and data values are repeated in pairs for each entry in the code table</p>
<i>Flag</i>	<p>32-bit flag. Low-order four hex digits are generic edit type; high-order four are styles within the type. A 1 in any bit indicates the corresponding style is checked. A 0 in any bit indicates the corresponding style is unchecked</p> <ul style="list-style-type: none"> <li>Bit 31: Sorted</li> <li>Bit 30: Allow editing</li> <li>Bit 29: Auto HScroll</li> <li>Bit 28: VScroll bar</li> <li>Bit 27: Always show list</li> <li>Bit 26: Always show arrow</li> <li>Bit 25: Uppercase</li> <li>Bit 24: Lowercase (if bits 25 and 24 are both 0, then case is Any)</li> <li>Bit 23: Empty string is NULL</li> <li>Bit 22: Required field</li> <li>Bit 21: Not used (set to 0)</li> <li>Bit 20: Not used (set to 0)</li> <li>Bits 19 – 16 (1 hex digit): Not used (set to 0)</li> <li>Bits 15 – 4 (3 hex digits): Always 0 for DropDownListBox edit style</li> <li>Bit 3: Always 0 for DropDownListBox edit style</li> <li>Bit 2: Always 0 for DropDownListBox edit style</li> <li>Bit 1: Always 1 for DropDownListBox edit style</li> <li>Bit 0: Always 0 for DropDownListBox edit style</li> </ul>

### DropDownDataWindow edit style (code 88)

Here is a sample row in the PBCatEdt table for a DropDownDataWindow edit style:

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>DataWin</i>	88	1	1	<i>Flag</i>	<i>Limit</i>
MyEdit	<i>DataCol</i>	88	1	2	0	<i>Key</i>
MyEdit	<i>DisplayCol</i>	88	1	3	0	<i>Width%</i>

where:

Value	Meaning
<i>DataWin</i>	Name of DataWindow object (report) to use
<i>DataCol</i>	Data column from DataWindow object (report)
<i>DisplayCol</i>	Display column from DataWindow object (report)
<i>Limit</i>	Character representation (in decimal) of <i>Limit</i> value
<i>Key</i>	One-character accelerator key
<i>Width%</i>	Width of the dropdown part of the DropDownDataWindow in %
<i>Flag</i>	<p>32-bit flag. Low-order four hex digits are generic edit type; high-order four are styles within the type. A 1 in any bit indicates the corresponding style is checked. A 0 in any bit indicates the corresponding style is unchecked</p> <ul style="list-style-type: none"> <li>Bit 31: Allow editing</li> <li>Bit 30: Auto HScroll</li> <li>Bit 29: VScroll bar</li> <li>Bit 28: Always show list</li> <li>Bit 27: Uppercase</li> <li>Bit 26: Lowercase (if bits 27 and 26 are both 0, then case is Any)</li> <li>Bit 25: HScroll bar</li> <li>Bit 24: Split horizontal scroll bar</li> <li>Bit 23: Empty string is NULL</li> <li>Bit 22: Required field</li> <li>Bit 21: Always show arrow</li> <li>Bit 20: Not used (set to 0)</li> <li>Bits 19 – 16 (1 hex digit): Not used (set to 0)</li> <li>Bits 15 – 8 (2 hex digits): Always 0 for DropDownDataWindow edit style</li> <li>Bit 7: Always 0 for DropDownDataWindow edit style</li> <li>Bit 6: Always 0 for DropDownDataWindow edit style</li> <li>Bit 5: Always 0 for DropDownDataWindow edit style</li> <li>Bit 4: Always 1 for DropDownDataWindow edit style</li> <li>Bit 3 – 0 (1 hex digit): Always 0 for DropDownDataWindow edit style</li> </ul>



## Edit edit style (code 89)

Here is a sample row in the PBCatEdt table for an Edit edit style:

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>Limit</i>	89	1	1	<i>Flag</i>	<i>Key</i>
MyEdit	<i>Format</i>	89	1	2	0	<i>Focus</i>
MyEdit	<i>Display1</i>	89	1	3	0	
MyEdit	<i>Data1</i>	89	1	4	0	
MyEdit	<i>Display2</i>	89	1	5	0	
MyEdit	<i>Data2</i>	89	1	6	0	

### About the example

This example shows an Edit edit style using a code table of display and data values. There is a pair of rows in PBCatEdt for each entry in the code table *only if* bit 23 of *Flag* is 1.

FOR INFO For information about code tables in edit styles, see Chapter 6, "Displaying and Validating Data".

where:

Value	Meaning
<i>Limit</i>	Character representation (in decimal) of <i>Limit</i> value
<i>Key</i>	One-character accelerator key
<i>Format</i>	Display format mask
<i>Focus</i>	Character "1" if Show Focus Rectangle is checked. NULL otherwise

Value	Meaning
<i>Flag</i>	<p>32-bit flag. Low-order four hex digits are generic edit type; high-order four are styles within the type. A 1 in any bit indicates the corresponding style is checked. A 0 in any bit indicates the corresponding style is unchecked</p> <p>Bit 31: Uppercase            Bit 30: Lowercase (if Bits 31 and 30 are both 0, then case is Any)            Bit 29: Auto selection            Bit 28: Password            Bit 27: Auto HScroll            Bit 26: Auto VScroll            Bit 25: HScroll bar            Bit 24: VScroll bar            Bit 23: Use code table            Bit 22: Validate using code table            Bit 21: Display only            Bit 20: Empty string is NULL            Bit 19: Required field            Bit 18: Not used (set to 0)            Bit 17: Not used (set to 0)            Bit 16: Not used (set to 0)            Bits 15 – 4 (3 hex digits): Always 0 for Edit edit style            Bit 3: Always 0 for Edit edit style            Bit 2: Always 0 for Edit edit style            Bit 1: Always 0 for Edit edit style            Bit 0: Always 1 for Edit edit style</p>

### Edit Mask edit style (code 90)

Here is a sample row in the PBCatEdt table for an Edit Mask edit style:

Name	Edit	Type	Cntr	Seqn	Flag	Work
MyEdit	<i>Format</i>	90	1	1	<i>Flag</i>	<i>DtFcKy</i>
MyEdit	<i>Range</i>	90	1	2	0	<i>SpinInc</i>
MyEdit	<i>Display1</i>	90	1	3	0	
MyEdit	<i>Data1</i>	90	1	4	0	
MyEdit	<i>Display2</i>	90	1	5	0	
MyEdit	<i>Data2</i>	90	1	6	0	

**About the example**

This example shows an Edit Mask edit style using a code table of display and data values as part of a spin control. Rows 2 and beyond exist in PBCatEdt only if the edit mask is defined as a spin control (bit 29 of *Flag* is 1). Rows 3 and beyond exist only if the optional code table is populated.

FOR INFO For information about using an edit mask as a spin control, see Chapter 6, "Displaying and Validating Data".

where:

<b>Value</b>	<b>Meaning</b>
<i>Format</i>	Display format mask
<i>DtFcKy</i>	Concatenated string with 1-character data-type code, 1-character focus-rectangle code (0 or 1), and 1-character accelerator key Data type codes: Format String = "0" Format Number = "1" Format Date = "2" Format Time = "3" Format DateTime = "4" Examples: "10x" means format is Number type, focus rectangle option is unchecked, accelerator key is "x" "31z" means format is Time type, focus rectangle option is checked, accelerator key is "z"
<i>Range</i>	Character representation (in decimal) of spin control range. The min value and max value are tab-delimited Example: "1[tab]13" means min = 1, max = 13
<i>SpinInc</i>	Character representation (in decimal) of spin increment
<i>Display1</i>	Display value for first entry in code table
<i>Data1</i>	Data value for first entry in code table
<i>Display2</i>	Display value for second entry in code table

<b>Value</b>	<b>Meaning</b>
<i>Data2</i>	<p>Data value for second entry in code table</p> <hr/> <p><b>About the display and data values</b>                      Display and data values are repeated in pairs for each entry in the code table</p> <hr/>
<i>Flag</i>	<p>32-bit flag. Low-order four hex digits are generic edit type; high-order four are styles within the type. A 1 in any bit indicates the corresponding style is checked. A 0 in any bit indicates the corresponding style is unchecked</p> <ul style="list-style-type: none"> <li>Bit 31: Required</li> <li>Bit 30: Autoskip</li> <li>Bit 29: Spin control</li> <li>Bit 28: Read only (code table option)</li> <li>Bit 27: Use code table</li> <li>Bit 26: Not used (set to 0)</li> <li>Bit 25: Not used (set to 0)</li> <li>Bit 24: Not used (set to 0)</li> <li>Bit 23 – 16 (2 hex digits): Not used (set to 0)</li> <li>Bit 15 – 8 (2 hex digits): Always 0 for Edit Mask edit style</li> <li>Bit 7: Always 0 for Edit Mask edit style</li> <li>Bit 6: Always 0 for Edit Mask edit style</li> <li>Bit 5: Always 1 for Edit Mask edit style</li> <li>Bit 4: Always 0 for Edit Mask edit style</li> <li>Bits 3 – 0 (1 hex digit): Always 0 for Edit Mask edit style</li> </ul>

# Index

## Symbols

- + operator 250, 598
- @
  - used in crosstabs 440
  - used in validation rules 295

## Numerics

- 24-hour times 274

## A

- Abs function 718
- absolute value 718
- accelerator keys
  - and CheckBox edit style 282
  - and RadioButton edit style 283
- AccuFonts for Windows
  - about 681
  - point sizes 682
- actions
  - assigning to buttons in forms 602
  - assigning to buttons in reports 254
  - in forms 529, 550
- aggregate functions
  - Avg 720
  - Count 729
  - CrosstabMax 737
  - CrosstabMin 739
  - CrosstabSum 741
  - CumulativePercent 743
  - CumulativeSum 745
  - First 758
  - in graphs 404
  - Large 773
  - Last 776
  - Max 789
  - Median 791
  - Min 795
  - Mode 799
  - Percent 809
  - restrictions 706, 710
  - Small 834
  - StDev 839
  - StDevP 842
  - Sum 847
  - Var 855
  - VarP 858
- alignment
  - in forms 578
  - in reports 238
- alignment extended attribute 92
- alignment for paragraphs 464
- Alter Table dialog box
  - altering table definition 97
  - applying display formats 267
  - applying edit styles 279
  - Font button 98
  - specifying extended attributes 93
- ALTER TABLE statement 105
- AND operator 163, 698
- angle
  - calculating cosine 728
  - calculating sine 833
  - calculating tangent 849
- appending 822
- AppleScript, executing scripts 35
- applications
  - about 614
  - creating 616
  - defining toolbars 619
  - distributing 615
  - executing pipelines 630
  - identifying in Help>About 625
  - initialization files for 621

- modifying commit value 635
- modifying maximum errors value 636
- modifying pipeline definition 633
- modifying pipeline type 635
- on different platforms 676
- pipng extended attributes 636
- prototyping 618
- repairing pipeline execution errors 633
- reusing most recent 622
- running 614, 624
- specifying icon for 618
- using pipelines 630
- using query governor 628
- working in Environment painter 614
- area graphs
  - description 392
  - making three-dimensional 395
- arguments
  - using in pipelines 504
  - using retrieval 173
- arithmetic operators 695
- arrays, in retrieval arguments 173
- Asc function 719
- ASCII text and rich text 458
- ASCII values, converting characters to 719
- asterisk in text patterns 787
- asterisks (\*), displaying user input as 280
- Auto Size setting, in graphs 419
- autoincrement columns, in forms 557
- Autosize Height
  - bands 230
  - with nested reports 380
- average value
  - columns 720
  - Crosstabs 731
- average, computing 251
- Avg function 720
- axes
  - scaling 424
  - specifying line styles 427
  - specifying properties in graphs 423
  - specifying text properties 418
  - using major and minor divisions 425

## B

- Background Color dropdown toolbar
  - in Form painter 566
  - in Report painter 198
- background colors
  - in forms 583
  - setting 221
- background.color property, on the Expressions page
  - about 344
  - specifying colors 364
- backslash in text patterns 786
- bands
  - in Report painter workspace 195
  - resizing in Report painter 204
  - setting colors for 221
- bar graphs
  - description 392
  - making three-dimensional 395
  - specifying overlap and spacing 422
- base reports 369
- BAT files 33
- BETWEEN operator 697
- bind variables, used with nested reports 383
- Bitmap function
  - about 723
  - example 714
- bitmap image files 674
- bitmaps
  - in rich text 467
  - specifying column as 85
- blob columns
  - creating 489
  - making visible 493
- blob data, creating columns for 489
- blobs
  - adding to reports 255
  - concatenating 700
- BMP files
  - adding to forms 599
  - adding to reports 246
  - in rich text 467
- BMP image format 674, 684
- boolean expressions
  - in filters 302
  - in validation rules 295, 298
- border property, on the Expressions page 345

- borders
    - around objects 589
    - defaults in forms 571
    - in forms 572
    - in graphs 427
    - in reports 229
  - Borders dropdown toolbar
    - in Form painter 566
    - in Report painter 198
  - boundaries, displaying in Report painter 235
  - brackets in text patterns 787
  - breaks, in grouped reports 308
  - brush.color property, on the Expressions page
    - about 346
    - specifying colors 364
  - brush.hatch property, on the Expressions page 346
  - business cards 148
  - buttons
    - adding to forms 600, 603
    - adding to reports 253
    - adding to toolbars 26
    - custom 29
    - deleting from toolbar 27
    - moving on toolbar 27
- C**
- caching data
    - in InfoMaker 207
    - turning off 208
  - cancel buttons in forms 602
  - capitalization
    - first letter 861
    - lowercase 785
    - uppercase 854
  - caret in text patterns 786
  - case
    - converting in reports and forms 280
    - sensitivity and code tables 290
  - Case function
    - about 724
    - example 714
  - categories, graph
    - basics 391
    - specifying 403
  - Category axis, graph 391
  - Ceiling function 726
  - century 862
  - Char function 727
  - characters
    - case of 719
    - changing capitalization 785, 854, 861
    - converting to ASCII values 719
    - extracting 794
    - matching 786
    - returning leftmost 778
    - returning rightmost 825
  - CheckBox edit style, defining 282
  - clipboard
    - copying data to 219
    - using 34
  - code tables
    - defining 288
    - description 288
    - in Specify Retrieval Criteria dialog box 243
    - processing 290
    - using display values in crosstabs 438
    - using display values in graphs 403
    - using in dropdown listboxes 281
  - color property, on the Expressions page
    - about 349
    - specifying colors 364
  - colors
    - calculating the numeric value 714
    - changing in Database painter 78, 103
    - defaults in forms 571
    - defining your own 584
    - for forms 572, 583
    - for reports 221
    - in display formats 269
    - in Select painter 168
    - red, green, and blue components of 823
    - table of standard colors 823
  - Column Display Format dialog box 266
  - column graphs
    - description 392
    - specifying overlap and spacing 422
  - column names 228
  - Column Object Validation page 297
  - Column Validation dialog box 294

- columns
  - adding new data source to forms 593
  - adding to forms 594
  - adding to reports 244
  - appending to table 95
  - applying display formats to 267
  - applying edit styles to 279
  - applying validation rules to 294
  - average value 720
  - checking for NULL value 708, 767
  - cumulative percent 743
  - cumulative sum 745
  - defining display formats 266, 267
  - defining edit styles 278
  - defining validation rules 293, 297
  - display value 784
  - displaying as a dropdown DataWindow 285
  - displaying as checkboxes 282
  - displaying as dropdown listboxes 281
  - displaying as radio buttons 283
  - displaying with fixed formats 283
  - first value 758
  - foreign key 106
  - formatting in reports 264
  - graphing data in 401
  - initial values 297
  - large value 773
  - last value 776
  - maximum value 789
  - median value 791
  - minimum value 795
  - most frequently occurring value 799
  - named in Report painter workspace 197
  - number of rows 729
  - percent of range 809
  - presenting in reports 276
  - preventing updates in forms 555, 558
  - removing display formats 267
  - removing edit styles 279
  - removing validation rules 295
  - reordering in grid forms 532, 576
  - reordering in grid reports 219
  - resizing in grid forms 532
  - resizing in grid reports 218
  - restricting input 283
  - selecting in Select painter 169
  - sliding to remove blank space 240
  - small value 834
  - specifying extended attributes 92
  - specifying for crosstabs 438
  - standard deviation 839, 842
  - total of values 708, 710, 847
  - updatable, in forms 555, 558
  - validating input in forms 292
  - validation rule of 761
  - value in code table 784
  - variable length 229
  - variance 855, 858
- comment extended attribute 92
- comments
  - displaying in Environment painter 49
  - including in SQL statements 128
  - modifying in Environment painter 50
- comparing strings 699
- comparison 764
- Composite presentation style
  - description 368
  - limitations 371
  - using 372
- composite reports
  - creating 372
  - description 368
  - limitations 371
  - specifying footer position 387
  - starting on new page 387
- computed columns, including in SQL Select 169
- computed field expressions 706
- computed fields
  - adding to forms 595, 597
  - adding to reports 248
  - creating from toolbar 30
  - defining 249
  - defining custom buttons for 252
  - in crosstabs 447
  - specifying display formats 267
  - summary statistics 251
- concatenation operator 700
- conditional expressions
  - example 710, 712, 716
  - IF function 763



- conditional modification
    - example, gray bar 332
    - example, highlighting rows 337
    - example, rotating objects 335
    - example, size and location 340
    - in forms 608
    - modifying objects 330
  - configuration settings, reading 814, 816
  - configuring ODBC on Windows 642
  - continuous data, graphing 392
  - Controls dropdown toolbar 566
  - copying objects in libraries 55
  - Cos function 728
  - cosine 728
  - count
    - computing 251
    - in graphs 404
  - Count function
    - about 729
    - example 708
  - count of values
    - columns 729
    - crosstabs 735
    - example 708
  - counting data, use of functions 710
  - counting NULLs, use of functions 708
  - Create Database command 79
  - Create Executable dialog box 617
  - Create Index dialog box 109
  - Create Local Database dialog box 79
  - Create Table dialog box
    - Font button 98
    - in Database painter 89, 90
    - specifying extended attributes 93
  - CREATE TABLE statement 105
  - CREATE VIEW statement 114
  - cross products 61
  - Crosstab Definition dialog box 437
  - CrosstabAvg function 731
  - CrosstabCount function 735
  - CrosstabMax function 737
  - CrosstabMin function 739
  - crosstabs
    - associating data 437
    - basic properties 444
    - changing column and row labels 446
    - changing definition of 445
    - creating 436
    - defining summary statistics 447
    - dynamic 434
    - functions 449
    - grid lines in 444
    - modifying data 445
    - overview 432
    - previewing 443
    - property conditional expressions in 454
    - specifying columns 438
    - specifying multiple columns and rows 441
    - static 434, 453
    - using expressions 439
    - using ranges of values 450
  - CrosstabSum function 741
  - CumulativePercent function 743
  - CumulativeSum function 745
  - CUR files, selecting mouse pointers 223, 586
  - CUR image format 674, 684
  - currency display format 268
  - currency, and rows 747, 760
  - current library
    - PBL file 46, 47, 48
    - setting 53
    - when opening InfoMaker 47
  - current row, finding the number of 713
  - CurrentRow function
    - about 747
    - example 713
  - cursors, image files 674
  - custom colors 584
  - Customize dialog box, description 27
- ## D
- data
    - accessing in freeform form 530
    - accessing in grid form 531
    - accessing in master/detail many-to-one form 534
    - accessing in master/detail one-to-many form 532
    - associating with graphs in reports 401
    - caching in InfoMaker 207
    - changing 119
    - converting to type long 783

- copying to clipboard 219
- data sources used in InfoMaker 155
- formatting in reports 264
- importing 123, 181, 209, 548
- limiting in forms 547
- pipng 500
- presenting in Freeform style 146
- presenting in Grid style 147
- presenting in Label style 147
- presenting in N-Up style 149
- presenting in reports 276
- presenting in Tabular style 146
- printing in forms 550
- retrieval options 61
- retrieving and updating 119, 259
- retrieving in reports 206, 208
- saving in external files 124, 212, 549
- saving in HTML Table format 213, 549
- selection options 61
- storing in reports 257
- updating, controlling 555
- validating in forms 292
- Data Manipulation painter
  - closing 125
  - description 70
  - opening 118
  - printing 124
  - sorting rows 120
- Data Pipeline painter
  - about 500
  - description 70
  - opening 52
  - working in workspace 506
- Data Retained On Save dialog box 257
- data source
  - configuration 656
  - defining for report 155
  - deploying 655
  - deploying on Macintosh 663
  - deploying on Windows 642
  - installing 655
  - installing on Macintosh 663
  - installing on Windows 642
  - modifying 232
  - modifying on Windows 645
  - queries on network 58
  - Query 180
  - Quick Select 156
  - SQL Anywhere 655, 656
  - SQL Select 166
  - Stored Procedure 182
- data type checking and conversion functions
  - Asc 719
  - Char 727
  - Date 748
  - DateTime 749
  - Integer 765
  - IsDate 766
  - IsNull 708, 767
  - IsNumber 768
  - IsTime 772
  - Long 783
  - Number 804
  - Real 819
  - String 845
  - Time 850
- data types
  - in display formats 269
  - in graphs 424
  - of blob columns 489
  - real 819
  - string 845
  - time 850
  - when piping data 501
- data validation
  - in code tables 290
  - with validation rules 292
- data values
  - in graphs 404
  - of code tables 288
  - specifying fonts in tables 98
  - using in graphs 403
- database administration
  - database access 126
  - executing SQL 128
  - painting SQL 128
  - security 126
- Database Administration painter
  - about 126
  - button 126
  - description 70
  - using 126

- Database Blob Object dialog box 490
- Database painter
  - accessing Database Administration painter 126
  - changing colors in 78
  - creating tables 88
  - defining display formats 266
  - defining validation rules 293
  - description 69
  - opening 72
  - previewing data 118
  - refreshing table list 81
  - specifying extended attributes 92
  - working with edit styles 278
  - workspace 72
- Database painter, validation rules 694
- database profiles
  - sharing in multiplatform environments 673
  - in pipelines 503
- databases
  - accessing through Quick Select 156
  - accessing through SQL Select 166
  - administering 126
  - changing 72
  - connecting to 143
  - controlling access to 126
  - controlling updates to 555
  - creating and deleting local SQL Anywhere 79
  - creating tables 88
  - destination in pipelines 500
  - ensuring referential integrity 102
  - executing SQL statements 131
  - extended attributes 683
  - importing data 123, 209, 548
  - limiting retrieved data 302
  - pipng data 500
  - retrieving, presenting, and manipulating data 118
  - source in pipelines 500
  - specifying fonts 98
  - stored procedures 182
  - system tables 82
  - updating 119
  - using as data source in a report 155
- DataWindow object 142
- Date function 748
- date, day, and time functions
  - Day 750
  - DayName 751
  - DayNumber 752
  - DaysAfter 753
  - Hour 762
  - Minute 797
  - Month 802
  - Now 803
  - RelativeDate 820
  - RelativeTime 821
  - Second 830
  - SecondsAfter 831
  - Today 851
  - Year 862
- dates
  - checking string 766
  - converting to 748
  - DateTime data type 749
  - day of week 751, 752
  - determining interval 753
  - display formats for 272
  - displaying in Environment painter 49
  - including in reports 253
  - obtaining current 851
  - obtaining day of month 750
- DateTime function 749
- Day function 750
- DayName function 751
- DayNumber function 752
- DaysAfter function 753
- DBMS
  - changing 72
  - controlling database access 126
  - CREATE VIEW statement 114
  - defining primary keys 105
  - executing SQL statements 131
  - generating SQL statement 117
  - specifying an outer join 116
  - stored procedures 182
  - supported 69
- defaults
  - buttons in forms 602
  - colors and borders in forms 571
  - for reports 153, 220
  - in forms 544
- Defining retrieval arguments 173
- Delete Database command 80

- DELETE statements
    - building in Database Administration painter 129
    - specifying WHERE clause 558
  - deleting, objects in Environment painter 56
  - deployment
    - on Macintosh 659
    - on Windows 637
  - Describe Rows dialog box 123
  - design, for multiple platforms 678
  - destination databases 500
  - display expressions in graphs 421
  - Display Format Definition dialog box 266
  - display formats
    - adding buttons in Report painter 268
    - applying to columns 267
    - applying to string 267
    - assigning from toolbar 30
    - colors in 269
    - data types 269
    - defining 269
    - defining in Report painter 267
    - deleting 300
    - description 264
    - for dates 272
    - for numbers 270
    - for strings 271
    - for times 273
    - in databases 92
    - in forms 610
    - in reports 262
    - maintaining 300
    - masks 269
    - removing 267
    - sections 269
    - using in graphs 420
    - working with in Database painter 266
  - display values
    - of code tables 288
    - using in crosstabs 438
    - using in graphs 403
  - displayed value from code table 784
  - display-only fields in forms 280
  - DISTINCT keyword 167
  - dollar sign in text patterns 786
  - drawing objects
    - adding to reports 245
    - setting color of 823
  - drop lines, graph 426
  - DROP VIEW statement 117
  - dropdown toolbars 24
  - DropDownDataWindow edit style
    - defining 285
    - defining code tables with 289
  - DropDownListBox edit style
    - defining 281
    - defining code tables with 289
  - duplicate values, index 109
- E**
- edges, displaying in Report painter 235
  - edit control, obtaining value in 761
  - Edit edit style
    - defining 280
    - defining code tables with 288
  - Edit Mask edit style
    - defining 283
    - defining code tables with 289
    - spin controls 285
  - Edit Style dialog box 278
  - edit styles
    - and selection criteria 161
    - applying to columns 279
    - deleting 300
    - description 276
    - in databases 92
    - in forms 610
    - in reports 262
    - in Specify Retrieval Criteria dialog box 243, 591
    - maintaining 300
    - removing 279
    - working with in Database painter 278
    - working with in Report painter 279
  - EditorExportFileType variable 677
  - elevation, in 3D graphs 428
  - Environment painter
    - about 46
    - comments, displaying 49
    - controlling display in 49
    - dates, displaying 49

- refreshing library list 49
    - what you can do in 46
    - workspace 48
  - Equality Required command 591
  - Equality Required property 243
  - error messages, customizing in validation rules 296
  - error rows, correcting in pipelines 516
  - executable files
    - about 614
    - creating 616
    - defining toolbars 619
    - naming 617
    - running 30, 624
  - Executable Items dialog box 619
  - Exp function 755
  - Explain SQL command 130
  - exponent 755
  - expressions
    - about 692
    - checking for NULL 708, 767
    - conditional evaluation 763
    - examples 692
    - in computed fields 250, 598
    - in crosstabs 439
    - in filters 302
    - in graphs 421
    - in OLE client names 492
    - in validation rules 295, 298
    - operators 695
    - specifying graph series with 405
    - specifying graph values with 404
  - Expressions property page 328
  - extended attributes
    - for fonts 586
    - pipng 502
    - used for text 586
  - extended column attributes
    - about 92
    - how stored 866
    - picture columns 85
    - used for text 227
  - External data source
    - importing data values 181
    - modifying result sets 233
  - external data, importing 123
  - external files
    - importing data from 209, 548
    - saving data in 212, 549
    - saving table data in 124
- ## F
- Fact function 756
  - file editor 33
  - files
    - names 671
    - ODBC, on Windows 652
    - on shared drives 671
    - rich text 462
    - sharing 670
  - Fill function 757
  - filters
    - expressions 706
    - in Data Manipulation painter 122
    - removing 303
  - First function 758
  - focus, moving from column to column 587
  - Font dialog box
    - changing text properties in 228
    - specifying fonts for tables 98
  - font.escapement property, on the Expressions page 349
  - font.height property, on the Expressions page 352
  - font.italic property, on the Expressions page 353
  - font.strikethrough property, on the Expressions page 354
  - font.underline property, on the Expressions page 355
  - font.weight property, on the Expressions page 355
  - fonts
    - changing in forms 587
    - choosing 678
    - database extended attributes 683
    - in reports, changing 227
    - PostScript 678
    - rich text formatting 464
    - size 681
    - specifying for tables 98
    - substitution 679
    - TrueType 678
  - footer bands, in Report painter workspace 198

- Foreground Color dropdown toolbar
  - in Form painter 566
  - in Report painter 198
- Foreign Key Definition dialog box 106
- foreign keys
  - defining 105
  - description 102
  - displaying in Database painter 103
  - joining tables 115, 171
  - opening related tables 104
- Form painter
  - copying and pasting objects 577
  - cutting objects 577
  - default colors and borders 571
  - deleting objects 575
  - equalizing size 579
  - equalizing spacing 579
  - keyboard shortcuts 568
  - moving objects 576
  - opening 52
  - resizing objects 577
  - sliding objects 580
  - toolbars in 565
  - undoing changes 580
  - using popup menu 567
  - using property sheets 568
  - working in workspace 565
- Form painter, validation rules 693
- Form property sheet
  - background color 583
  - specifying a scrollbar 585
  - specifying a title 583
- form styles
  - about 529
  - style libraries 59
- format property, on the Expressions page 356
- forms 705
  - about 526
  - and the repository 544
  - accessing 553
  - actions 550
  - adding columns 594
  - adding computed fields 597
  - adding objects 594
  - aligning objects 578
  - borders in 589
  - buttons, adding 600, 603
  - computed fields, adding 595
  - controlling updates in 555
  - copying 55
  - creating basic form 536
  - creating master/detail many-to-one 537
  - creating master/detail one-to-many 537
  - creating new 528, 536
  - data source, modifying 592
  - data sources 528
  - default colors and borders 571
  - deleting 56, 553
  - design for multiple platforms 678
  - designing 676
  - distributing 614
  - drawing objects, adding 606
  - enhancing 564
  - fonts 678
  - freeform style 530
  - generating 543
  - grid style 531
  - highlighting information 608
  - how stored 53
  - in applications 527
  - inserting rows in database 543
  - limiting data 547
  - master/detail many-to-one style 529, 534, 537
  - master/detail one-to-many style 529, 532, 537
  - modifying objects 574
  - modifying properties 582
  - moving 55
  - naming 545
  - objects, position and size 571
  - opening 553
  - performance 672
  - pictures, adding 599
  - printing 550
  - printing definition 573
  - prompting for criteria 589
  - reports, adding 605
  - running 546
  - running in executable 627
  - saving 545
  - saving data in 549
  - selecting rows when running 589
  - setting borders 572

- setting colors 572, 583
  - tab order 587
  - text, adding 594
  - titles in 582
  - U.S. number format 706
  - updating database 542
  - using scrollbars 584
  - validation rules 292
  - why used 526
  - freeform forms 529
    - about 530
    - default colors and borders 571
  - freeform reports
    - about 146
    - default wrap height 154
    - detail band in 197
    - header band in 196
  - functionality, Macintosh and Windows 684
  - functions
    - aggregate 706, 710
    - crosstab 449
    - example, counting data 710
    - example, counting NULLs 708
    - example, displaying data 715
    - example, row indicator 713
- G**
- General keyword, in number display formats 271
  - GetRow function
    - about 760
    - example 713
  - GetText function
    - about 761
    - using in validation rules 298
  - Graph Object property sheet 401, 415
  - graphics
    - adding to forms 599
    - adding to reports 246
    - formats 684
  - graphs
    - adding to reports 138, 255
    - autosizing text 419
    - changing position of 399
    - data types of axes 424
    - default positioning in reports 256
    - defining properties 415
    - examples 406
    - expressions in 421
    - in reports 396
    - legends in 417
    - major and minor divisions 425
    - multiple series 405
    - names of 416
    - overlays in 412
    - overview 390
    - parts of 390
    - property sheets 398
    - rotating text 420
    - scaling axes 424
    - selecting data 401
    - single series 404
    - sorting series and categories 417
    - specifying borders 427
    - specifying categories 403
    - specifying overlap and spacing of
      - bars/columns 422
    - specifying pointers 428
    - specifying properties of axes 423
    - specifying rows 402
    - specifying series 404
    - specifying the type 417
    - specifying values 404
    - text properties in 418
    - titles in 416
    - types of 392
    - using display formats 420
    - using Graph presentation style 414
  - grid
    - aligning objects in forms 574
    - aligning in reports 236
  - grid forms 529
    - about 531
    - default colors and borders 571
    - reordering columns 532, 576
    - resizing columns 532, 577
  - grid lines, graph 426
  - grid reports
    - about 147
    - basic properties 222
    - detail band in 197

- displaying grid lines 222
- header band in 196
- reordering columns 219
- resizing columns 218
- working in 218
- GROUP BY criteria 178
- Group presentation style
  - properties of 313
  - using 310
- groupbox, adding to reports 246
- grouping
  - in SQL Select 178
  - restricting 179
- groups in reports
  - graphing 402
  - of rows 308
  - sorting 322

## H

- HAVING criteria 179
- heading extended attribute 92
- headings
  - in reports 196
  - specifying fonts in tables 98
- height property, on the Expressions page 357
- help, using 18
- highlighting information in forms 608
- highlighting, rows 771
- hour 762
- Hour function 762
- HTML Table format, saving data in 213, 549

## I

- ICO image format 674, 684
- icons
  - image files 674
  - specifying for executables 618
- identity columns, in forms 557
- If function
  - about 763
  - example 710, 712, 716

- IM.INI files
  - file format 39
  - how InfoMaker finds them 39
  - sharing 62
  - variable for setting line endings 677
- image files, supported 674, 684
- image, in computed field 723
- importing data 123, 209, 548
- IN operator
  - about 697
  - in Quick Select 163
- indexes
  - creating 109
  - dropping from tables 107, 110, 111
- InfoMaker
  - environment, managing 46
  - sharing files 670
  - supported functionality 684
- InfoMaker expressions 692, 705
- InfoMaker functions
  - Abs 718
  - Asc 719
  - Avg 720
  - Bitmap 723
  - Case 724
  - Ceiling 726
  - Char 727
  - Cos 728
  - Count 729
  - CrosstabAvg 731
  - CrosstabCount 735
  - CrosstabMax 737
  - CrosstabMin 739
  - CrosstabSum 741
  - CumulativePercent 743
  - CumulativeSum 745
  - CurrentRow 747
  - Date 748
  - DateTime 749
  - Day 750
  - DayName 751
  - DayNumber 752
  - DaysAfter 753
  - Exp 755
  - Fact 756
  - Fill 757



- First 758
- GetRow 760
- GetText 761
- Hour 762
- If 763
- Int 764
- Integer 765
- IsDate 766
- IsNull 767
- IsNumber 768
- IsRowModified 769
- IsRowNew 770
- IsSelected 771
- IsTime 772
- Large 773
- Last 776
- Left 778
- LeftTrim 779
- Len 780
- Log 781
- LogTen 782
- Long 783
- LookUpDisplay 784
- Lower 785
- Match 786
- Max 789
- Median 791
- Mid 794
- Min 795
- Minute 797
- Mod 798
- Mode 799
- Month 802
- Now 803
- Number 804
- Page 805
- PageAcross 806
- PageCount 807
- PageCountAcross 808
- Percent 809
- Pi 812
- Pos 813
- ProfileInt 814
- ProfileString 816
- Rand 818
- Real 819
- RelativeDate 820
- RelativeTime 821
- Replace 822
- RGB 823
- Right 825
- RightTrim 826
- Round 827
- RowCount 828
- RowHeight 829
- Second 830
- SecondsAfter 831
- Sign 832
- Sin 833
- Small 834
- Space 837
- Sqrt 838
- StDev 839
- StDevP 842
- String 845
- Sum 847
- Tan 849
- Time 850
- Today 851
- Trim 852
- Truncate 853
- Upper 854
- Var 855
- VarP 858
- WordCap 861
- Year 862
- INI files *see* initialization files
- initial values, for columns 297
- initialization files 39, 40
  - editing 33
  - for executable files 621
  - how InfoMaker finds them 39
  - modifying to change pipeline definitions 633
  - modifying to identify applications 625
  - platform differences 673
  - reading 814, 816
  - sharing 673
- input fields
  - about 465
  - columns 465
  - computed fields 466
  - data 459

- data format 466
  - editing data 465
  - flashing 465
  - selected 465
  - validation rules 466
  - Input Validation dialog box 294
  - INSERT statements, building in Database
    - Administration painter 129
  - inserting strings 822
  - Int function 764
  - integer
    - converting to 765
    - converting to char 727
  - Integer function 765
  - IsDate function 766
  - IsNull function
    - about 767
    - example 708, 716
  - IsNumber function 768
  - IsRowModified function 769
  - IsRowNew function 770
  - IsSelected function 771
  - IsTime function 772
- J**
- Join dialog box 115
  - joins
    - in Select painter 171
    - number of tables in 61, 63
- K**
- key and modified columns, updating rows 558
  - key and updatable columns, updating rows 558
  - key columns for OLE columns 489
  - key modification, updating rows 561
  - keyboard
    - shortcuts in Database painter 77
    - shortcuts in Form painter 568
    - shortcuts in Report painter 202
  - keys, database
    - arrows specifying key relationship 158
    - displaying in Database painter 103
    - dropping from tables 107
    - specifying in forms 557
    - updating values in forms 561
    - using primary and foreign 102
  - keywords, display format 269
- L**
- label extended attribute 92
  - label style
    - detail band in 197
    - of reports 147
    - removing blank lines 240
    - specifying fonts in tables 98
  - Large function 773
  - Last function 776
  - Layout dropdown toolbar
    - in Form painter 566
    - in Report painter 198
  - Left function 778
  - LeftTrim function 779
  - legends
    - in graphs 391
    - specifying text properties 418
    - using 417
  - Len function 780
  - libraries 671
    - creating 54
    - for forms, reports, queries, pipelines 53
    - of form styles 59
    - of queries 58
    - PBL files 46, 47
    - refreshing library list 49
  - LIKE operator
    - about 697
    - in Quick Select 162
  - line drawing objects 245, 606
  - line endings 677
  - line graphs
    - description 392
    - making three-dimensional 395
  - line styles, graph 427
  - local SQL Anywhere databases 79
  - LOG file, SQL Anywhere 675
  - Log function 781

logarithms 781, 782  
 logical expressions, truth table 698  
 logical operators 162, 698  
 LogTen function 782  
 Long function 783  
 longs, converting to 783  
 LookUpDisplay function 403, 784  
 Lower function 785  
 lowercase 785

## M

Macintosh  
   executing AppleScript 35  
   feature support 684  
   font substitutions 680  
   popup menus 26  
 mail and messaging 684  
 mailing labels 147  
 mailing reports (PSR files) 217  
 major divisions, in graphs 425  
 MAPI 684  
 masks  
   for display formats 269  
   matching 786  
 master/detail many-to-one forms  
   about 529, 534  
   creating 537  
 master/detail one-to-many forms  
   about 529, 532  
   creating 537  
 Match button with validation rules 296  
 Match function 786  
 match patterns, validation rules 296  
 Max function 789  
 maximum value  
   below a limit 764  
   columns 789  
   crosstabs 737  
 Median function 791  
 menu bars, in applications 625  
 metacharacters 786  
 metafiles, specifying columns as 85  
 Mid function 794  
 military time 274

Min function 795  
 minimum value  
   above a limit 726  
   columns 795  
   crosstabs 739  
 minor divisions, in graphs 425  
 Minute function 797  
 Mod function 798  
 Mode function 799  
 Modify Result Set Description dialog box 233  
 modulus 798  
 Month function 802  
 month, obtaining the day of 750  
 multiplatform applications  
   creation 676  
   design 678  
   fonts 678  
   performance 676  
 multiple-series graphs 405

## N

name tags 148  
 names  
   of columns in Report painter workspace 197  
   of executable files 617  
   of forms 545  
   of graphs 416  
   of objects in reports 228  
   of queries 189  
   of reports 187  
 negative numbers 832  
 nested reports  
   adding another report 382  
   adding to report (DataWindow) 141, 374  
   autosize height 380  
   changing 381  
   changing content of 382  
   description 368  
   displayed in workspace 377  
   how retrieval works 371  
   limitations 371  
   slide options 387  
   specifying criteria 385  
   using retrieval arguments 376, 383

- width, adjusting 380
  - New Form dialog box 528, 536
  - new lines in text 587
  - New Name dialog box 446
  - newline characters in text 227, 587
  - newspaper columns 225
  - NOT BETWEEN operator 697
  - NOT IN operator 697
  - NOT LIKE operator 697
  - NOT operator 698
  - Now function 803
  - NULL values
    - allowing in code tables 288
    - allowing in tables 90
    - altering table definition 96
    - and blob columns 489
    - checking 708, 367
    - ignored in aggregate 721, 729, 743, 790, 792, 796, 800, 810
    - in expressions 696
    - specifying display formats for 270
  - Number function 804
  - numbers
    - checking string 768
    - determining maximum 726
    - determining sign of 832
    - display formats for 270
    - logarithm of 781, 782
    - multiplying by pi 812
    - of day of week 752
    - random 818
    - returning remainder 798
    - rounding 827
    - truncating 853
    - U.S. format 706
  - numeric functions
    - Abs 718
    - Ceiling 726
    - Cos 728
    - Exp 755
    - Fact 756
    - Int 764
    - Log 781
    - Mod 798
    - Pi 812
    - Rand 818
    - Round 827
    - Sign 832
    - Sin 833
    - Sqrt 838
    - Tan 849
    - Truncate 853
  - N-Up style
    - computed fields in 250
    - detail band in 197
    - header band in 196
    - of reports 149
- O**
- object boundaries 235
  - object libraries 671
  - object linking and embedding 684
  - objects
    - adding to forms 594
    - adding to reports 244
    - aligning in forms 578
    - aligning in Report painter 238
    - copying in forms 577
    - copying in Report painter 237
    - cutting in forms 577
    - deleting from forms 575
    - deleting from Report painter 236
    - equalizing size 239, 579
    - equalizing spacing 239, 579
    - modifying in forms 574
    - moving in forms 576
    - moving in Report painter 237
    - pasting in forms 577
    - position and size 571
    - resizing in forms 577
    - resizing in Report painter 238
    - selecting in forms 570
    - selecting, in Report painter 200
    - sliding in forms 580
    - undoing changes 580
  - Objects dropdown toolbar, in Report painter 198
  - ODBC driver 642
  - ODBC.INI 642
  - ODBCINST.INI 642

- OLE
    - blob column 672
    - columns in reports 489
    - functionality 684
    - previewing columns 493
    - report objects 476
    - Report presentation style 478
  - OLE Database Blob command 490
  - OLE object
    - activating object 487
    - display of 488
    - embedding 487
    - icon for 488
    - linking 487
    - updating link 488
  - operators
    - arithmetic 695
    - concatenation 700
    - in Quick Select criteria 162
    - logical 698
    - precedence 701
    - relational 696
    - used in expressions 695
  - options, choosing for reports 153
  - Options property sheet
    - specifying colors and borders for forms 572
    - using the grid in forms 575
  - OR operator
    - about 698
    - in Quick Select 163
  - ORDER BY clause
    - in SELECT statements 177
    - specifying in Quick Select 160
  - outer joins
    - definition 61
    - specifying 116
  - oval drawing objects 245, 606
  - overlap, of columns in graphs 422
  - overlays, in graphs 412
  - Override Edit command 591
- P**
- page
    - current 805
    - current horizontal 806
    - total 807
    - total across 808
  - Page function 805
  - page numbers, including in reports 253
  - PageAcross function 806
  - PageCount function 807
  - PageCountAcross function 808
  - PainterBar
    - adding custom buttons to 29
    - controlling display of 25
    - description 24
  - painters, opening 16
  - painting SQL statements 128
  - paragraph alignment 464
  - parsing strings 778, 813
  - passwords
    - displaying as asterisks 280
    - fields 280
  - pasting
    - objects in forms 577
    - SQL statements, in Database Administration painter 128
  - pattern matching 786
  - PBCatCol system table 82, 869
  - PBCatEdt system table 82, 872
  - PBCatFmt system table 82, 870
  - PBCatTbl system table 82, 867
  - PBCatVld system table 82, 871
  - PBL files
    - about 53
    - copying and moving 672
    - definition of 47
    - fonts for multiple platforms 682
    - on shared drives 671
    - query libraries 58
    - style libraries 59
    - text for multiple platforms 681
    - TUTOR\_IM.PBL 47
    - Windows to Macintosh 672
  - PBLAB60.INI 149
  - pen.color property, on the Expressions page
    - about 357
    - specifying colors 364
  - pen.style property, on the Expressions page 358
  - pen.width property, on the Expressions page 359

- percent display format 268
- Percent function 809
- performance improvement 672, 676
- period in text patterns 786
- periodic data, in reports 149
- perspective, in 3D graphs 428
- phone lists, creating 225
- Pi function 812
- PICT image format 674, 684
- picture buttons, adding to forms 603
- pictures
  - adding to forms 599
  - adding to reports 246
  - specifying column as 85
- pictures, in computed fields 714, 723
- pie graphs
  - description 393
  - making three-dimensional 395
- pipelines
  - about 500
  - copying 55
  - creating 503
  - data types, supported 501
  - deleting 56
  - destination database 500
  - destination, changing 514
  - distributing 614
  - editing source data 505
  - error messages 517
  - errors, correcting 517
  - examples 500, 520
  - executing 505
  - executing in applications 630
  - execution, stopping 511
  - extended attributes 502
  - how stored 53
  - modifying 506
  - modifying comments 50
  - modifying commit value in applications 635
  - modifying definition in applications 633
  - modifying maximum errors value in applications 636
  - modifying type in applications 635
  - moving 55
  - opening 519
  - pipeline operations 508
  - pipng extended attributes in applications 636
  - repairing execution errors in applications 633
  - retrieval arguments 504
  - reusing 519
  - rows, committing 511
  - running in executable 627
  - saving 518
  - source database 500
  - using in applications 630
- pixel size 683
- pixels, as report unit of measure 220
- placeholders, in validation rules 295
- plus sign in text patterns 787
- point of view, in 3D graphs 428
- point sizes 682
- pointer property, on the Expressions page 360
- pointers
  - image files 674
  - in forms 585
  - in graphs 428
  - in reports 222
- points, specifying size for tables 98
- popup menus
  - controlling toolbars with 25
  - in Form painter 567
  - on the Macintosh 26
  - using 20
- Pos function 813
- positive numbers 832
- PostScript fonts 678
- PowerBar
  - adding custom buttons to 29
  - controlling display of 25
  - description 24
  - displaying available buttons 27
  - using 14
- Powersoft Demo Database 47
- Powersoft Report (PSR) files 684
- PowerTips
  - assigning text in custom buttons 30
  - using 15
- precedence, operators 701
- previewing
  - displaying rulers 210
  - for crosstabs 443
  - Form painter 52
  - importing data 209

- in Report painter 205
- OLE report objects 482
- Query painter 52
- Report painter 52
- retrieving rows 206
- sorting and filtering data 209
- primary buffer 828
- Primary Key Definition dialog box 104
- primary keys
  - defining 104
  - description 102
  - displaying in Database painter 103
  - identifying updatable rows 557
  - joining tables 115, 171
  - opening related tables 104
- print specifications, reports 223
- printing
  - form definitions 573
  - forms 550
  - in Data Manipulation painter 124
  - reports 211
- profile files, reading 814, 816
- ProfileInt function 814
- profiles, in pipelines 503
- ProfileString function 816
- Prompt for Criteria dialog box 242, 590
- properties
  - about 326
  - defining for graphs 415
  - example, gray bar 332
  - example, highlighting rows 337
  - example, rotating objects 335
  - example, size and location 340
  - modifying objects 330
  - specifying colors 364
  - using expressions 328
- property conditional expressions 326, 454
- property sheets
  - about 22
  - arrangement of tabs 22
  - buttons 22
  - displaying 23
  - for graphs 398
  - in Form painter 568
  - in Report painter 199

- property values on the Expressions property page
  - about 343
  - background.color 344
  - border 345
  - brush.color 346
  - brush.hatch 346
  - color 349
  - font.escapement 349
  - font.height 352, 357
  - font.italic 353
  - font.strikethrough 354
  - font.underline 355
  - font.weight 355
  - format 356
  - pen.color 357
  - pen.style 358
  - pen.width 359
  - pointer 360
  - specifying colors 364
  - supplying 343
  - visible 360
  - width 361
  - x 362
  - x1, x2 362
  - y 363
  - y1, y2 363
- PSR files 684
  - creating 212
  - definition of 47
  - mailing 217
  - opening 216
  - overview 215

## Q

- queries
  - accessing on network 58
  - copying 55
  - defining 188
  - deleting 56
  - how stored 53
  - modifying 190
  - modifying comments 50
  - modifying SELECT statement graphically 592
  - moving 55

- naming 189
- previewing 188
- running from toolbar 30
- saving 189

Query data source 180

Query Governor

- about 61
- button 62
- defaults 62
- number of tables in joins 63
- running in application 628

query libraries 58

Query painter, opening 52, 188

question mark in text patterns 787

Quick Select

- defining data source as 156
- up and down arrows 158
- modifying SELECT statement graphically 592

## R

RadioButton edit style, defining 283

Rand function 818

random numbers, obtaining 818

ranges, cross-tabulating 450

Real function 819

Rebuild Columns At Runtime checkbox 453

rectangle drawing objects 245, 606

referential integrity, in databases 102

refreshing library list 49

RegEdit utility, adding OLE server applications 492

relational operators 696

RelativeDate function 820

RelativeTime function 821

remainder 798

Replace function 822

Report painter

- copying objects 237
- defining display formats 267
- defining validation rules 297
- deleting objects 236
- equalizing size 239
- equalizing spacing 239
- importing data 209
- keyboard shortcuts 202

MicroHelp 202

moving objects 237

nesting reports 368

opening 52

resizing bands 204

resizing objects 238

retrieving data 206

saving data 212

selecting objects 200

sliding objects 240

toolbars in 198

undoing changes 204

using property sheets 199

working in the workspace 195

working with edit styles 279

zooming objects 204

report, OLE

- OLE object 478, 479
- presentation style 478, 479
- previewing 482
- see also* OLE object

reports

- about the repository 185
- adding objects 244
- adding to forms 605
- aligning objects 238
- and graphs in 396
- blobs, adding 255
- borders in 229
- business cards 148
- buttons, adding 253
- caching data 207
- changing margins 210
- columns, adding 244
- Composite style 141, 368
- computed fields, adding 248
- computed fields, defining 249
- copying 55
- creating new 143
- creating OLE columns 489
- Crosstab style 140
- custom buttons that add computed fields 252
- data source, modifying 232
- data sources 155
- data, storing in 257
- defaults 153, 220



- deleting 56
- designing for multiple platforms 678
- display formats 264
- displaying boundaries 235
- distributing 614
- drawing objects, adding 245
- edit styles 276
- effect of Query Governor 208
- expressions in computed fields 250
- fonts 678
- Freeform style 139
- generating 185
- Graph style 414
- graphs, adding 138, 255
- Grid style 138, 222
- grid, working in 218
- Group style 137, 310
- groupbox, adding 246
- how stored 53
- InfoMaker expressions 705
- initial values for columns 297
- Label style 139
- mailing 217
- modifying 143
- modifying comments 50
- moving 55
- multiple column 149
- name tags 148
- naming 187
- nested reports 141
- newspaper columns in 225
- performance 672
- pictures, adding 246
- positioning of objects in 256
- presentation styles 145
- previewing 205
- previewing without retrieving data 207
- print specifications 223
- printing 211
- prompting for criteria 242
- PSR files 212
- repository information used 227
- resizing objects during execution 229
- result sets, modifying 233
- retrieval arguments, modifying 232
- retrieval criteria 242
  - retrieving as needed 259
  - retrieving data 206, 208
  - running from toolbar 30
  - running in executable 627
  - saving 186
  - setting colors 221
  - setting timer 221
  - sorting rows 305
  - suppressing repeating values 306, 308
  - Tabular style 136, 137
  - text, adding 245
  - turning off caching 208
  - U.S. number format 706
  - units of measure 220
  - years, how interpreted 272
- repository
  - about 82, 185, 544, 865
  - deleting orphan table information 86
  - information used in forms 544, 586
  - information used in reports 185, 227
  - piping 502
  - storing display formats 266
  - storing edit styles 278
  - storing extended attributes 93
  - storing validation rules 293
- resizing objects during execution 229
- result sets, modifying 233
- Retain Data to Design option 208
- retrieval arguments
  - defining 173
  - in nested reports 376
  - modifying in reports 232
  - specifying in pipelines 504
- retrieval criteria
  - in nested reports 385
  - in Quick Select grid 162
  - prompting for in forms 589
  - prompting for in reports 242
- Retrieve command 206
- Retrieve on Preview option 206
- Retrieve Only As Needed 259
- Retrieve Rows to Disk command 260
- retrieving data
  - effect of Query Governor 208
  - options 61
  - retrieve as needed 208

- RGB function
    - about 823
    - example 714
  - rich text 457
    - about 458
    - file for report 462
    - header and footer 463
    - pictures 467
    - tables 458
    - toolbars 460
    - unsupported formatting 458
  - RichText presentation style
    - about 459
    - columns 465
    - creating 460
    - editing keys 471
    - header and footer 460, 463, 469
    - objects 463
    - page numbers 463
    - paragraph settings 464
    - picture objects 467
    - preview 462, 469
    - print preview 469
    - protected from changes 460
    - report settings 464
    - Rich Text Object settings 464
    - selected text settings 464
    - setting up 461
    - settings 460
    - today's date 463
  - RichTextEdit controls, editing keys 471
  - Right function 825
  - RightTrim function 826
  - RLE files
    - adding to forms 599
    - adding to reports 246
  - rotation
    - in 3D graphs 428
    - of text in graphs 420
  - Round function 827
  - Round Maximum To, in graphs 424
  - RoundRectangle drawing objects 245, 606
  - Row indicator, use of functions 713
  - RowCount function 828
  - RowHeight function 829
  - rows
    - and bands 760
    - displaying information about 123
    - errors in pipelines 516
    - filtering 122, 302
    - getting current 713, 747, 760
    - graphing 402
    - grouping 178, 308
    - height 829
    - in primary buffer 828
    - modification status 769, 770
    - modifying in Data Manipulation painter 119
    - number retrieved 61
    - removing filters 303
    - retrieving 119
    - retrieving as needed 259
    - row with focus 747
    - saving in external files 124
    - selecting 771
    - selecting when running forms 589
    - sorting 120, 177, 305
    - suppressing repeating values 306
    - using forms to insert 543
  - RTF 457
  - rulers
    - in Report painter 236
    - displaying in print preview 210
  - running forms 546
- ## S
- Save As dialog box 212
  - Save Rows As dialog box 124
  - saving
    - data in external files 212, 549
    - data in HTML Table format 213, 549
    - data in reports 257
    - forms 545
    - pipelines 501, 518
    - queries 189
    - reports 186
  - scatter graphs, description 393
  - screen size and resolution 683
  - scroll bars 678
  - scrollbars in forms 584

- Second function 830
- SecondsAfter function 831
- Select Detail Table dialog box 539
- SELECT DISTINCT statements 61
- Select Form dialog box 536
- Select Icon dialog box 618
- Select Master Table dialog box 538
- Select Master/Detail relationship dialog box 540
- Select painter
  - adding tables 168
  - colors in 168
  - defining retrieval arguments 173
  - joining tables 171
  - opening 166
  - saving work as query 166
  - selecting tables 167
  - specifying selection, sorting, and grouping criteria 175
  - specifying what is displayed 168
  - SQL toolbox 167
- SELECT statements
  - building in Database Administration painter 129
  - displaying 170
  - editing syntactically 171
  - for view, displaying 114
  - limiting data retrieved 302
  - modifying in reports 592
  - predefined 188
  - saved as queries 188
  - sorting rows 305
- Select Tables dialog box 72, 81, 112
- selection criteria
  - allowing users to specify 242
  - specifying for forms 589
  - specifying in Quick Select 161
  - specifying in SQL Select 176
- Series axis, graph 391
- series, graph
  - as overlays 412
  - basics 390
  - specifying 404
- shared drives, files on 671
- shared IM.INI files 62
- SharedIni variable, multiplatform environments 673
- sharing files
  - image files 674
  - INI files 673
  - multiplatform environment 671
  - object libraries 671
  - text files 672
- shortcut keys
  - in Form painter 568
  - in Report painter 202
- Show Edges option 235
- Sign function 832
- Sin function 833
- sine 833
- single-series graphs 404
- size
  - equalizing in forms 579
  - equalizing in Report painter 239
  - of bands in Report painter 204
  - of form objects 577
  - of reports 238
  - undoing changes 580
- size, of string 780
- Slide dropdown toolbar,
  - in Report painter 198
- sliding objects
  - in forms 580
  - in nested reports 387
  - in reports 240
- Small function 834
- snaking columns in reports 225
- sort criteria, specifying in Quick Select 160
- sorting
  - groups 322
  - in graphs 417
  - in SQL Select 177
  - rows 305
- source databases 500
- sources of data 155
- Space function 837
- spaces
  - deleting leading 779
  - deleting trailing 826
  - inserting in a string 837
  - removing from strings 852
- spacing
  - equalizing in forms 579

- equalizing in Report painter 239
- of columns in graphs 422
- Specify Retrieval Criteria dialog box
  - displaying to users 242
  - when running forms 590
- Specify Sort Columns dialog box 305
- Specify Update Properties dialog box 556
- spin controls, defining edit masks as 285
- SQL Anywhere
  - configuration 656
  - databases, creating and deleting 79
  - data source 655, 656
  - Macintosh 668
  - runtime system 657
- SQL Select
  - adding tables 168
  - defining retrieval arguments 173
  - joining tables with 171
  - selecting columns 169
  - selecting tables 167
  - specifying selection, sorting, and grouping criteria 175
  - specifying what is displayed 168
  - using as data source 166
- SQL Select data source
  - colors in 168
  - modifying SELECT statement graphically 592
  - SQL toolbox 167
- SQL Statement Type dialog box 128
- SQL statements
  - building and executing 128
  - displaying 170
  - executing 128, 131
  - execution plan 130
  - explaining 130
  - for views, displaying 114
  - generating through Quick Select 156
  - generating through SQL Select 166
  - importing from text files 130
  - painting 128
  - typing 129
- SQL toolbox 167
- SQLCache variable 383
- Sqrt function 838
- square root 838
- stacked graphs 395
- standard deviation 839, 842
- Start on New Page command 387
- StDev function 839
- StDevP function 842
- Stored Procedure data source 182
- stored procedures
  - modifying result sets in reports 233
  - using 182
- String function 845
- string functions
  - Asc 719
  - Char 727
  - Fill 757
  - Left 778
  - LeftTrim 779
  - Len 780
  - Lower 785
  - Match 786
  - Mid 794
  - Pos 813
  - Replace 822
  - Right 825
  - RightTrim 826
  - Space 837
  - Trim 852
  - Upper 854
  - WordCap 861
- strings
  - comparing 699
  - concatenating 250, 598, 700
  - converting 748, 783, 804, 819
  - deleting leading spaces 779
  - detecting contents 766, 768, 772
  - display formats for 271
  - extracting 794
  - finding substrings 813
  - lowercase 785
  - uppercase 854
- Style dialog box, Report painter 220
- style libraries 59
- StyleBar
  - controlling display of 25
  - description 24
  - in Form painter 565, 567
  - in Report painter 199

- substring
    - extracting 794
    - finding 813
    - replacing 822
  - sum
    - computing 251
    - in graphs 404
  - Sum function
    - about 847
    - example 708, 710
  - summary bands, in Report painter workspace 198
  - summary statistics
    - computing 251
    - in crosstabs 447
  - syntax, of view SELECT statement 114
  - system and environment functions
    - ProfileInt 814
    - ProfileString 816
  - system date 851
  - system options tab 40
  - system tables
    - DBMS 82
    - PowerBuilder/InfoMaker 82, 93, 865
  - system time 803
- T**
- tab order in forms 587
  - Table painter
    - applying validation rules 294
    - description 70
  - tables
    - accessing in freeform form 530
    - accessing in grid form 531
    - accessing in master/detail many-to-one form 534
    - accessing in master/detail one-to-many form 532
    - altering definition of 95
    - applying display formats to columns 267
    - applying edit styles to columns 279
    - applying validation rules to columns 294
    - controlling updates to 555
    - creating 88
    - creating indexes 109
    - dropping 86
    - dropping indexes 107, 110, 111
    - extended attributes, specifying 92
    - joining in Select painter 171
    - joins, number of tables in 61
    - number in joins 61
    - opening in Database painter 81
    - opening, related to foreign keys 104
    - opening, related to primary keys 104
    - presenting in Freeform style 146
    - presenting in Grid style 147
    - presenting in Label style 147
    - presenting in N-Up style 149
    - presenting in Tabular style 146
    - printing data 124
    - refreshing list 81
    - removing from Database painter workspace 86
    - resizing in Database painter workspace 75
    - rich text 458
    - rows, number retrieved 61
    - saving data in external files 124
    - selecting for SQL Select 166
    - selecting, in foreign key definition 106
    - specifying fonts 98
    - specifying updatable 556
    - working with data 118
  - Tabular style
    - detail band in 197
    - header band in 196
    - of reports 146
  - Tan function 849
  - tangent 849
  - target data for OLE 484
  - testing values 714
  - text
    - cutting, copying, and pasting 34, 97
    - editing 33
    - finding substrings 813
    - in forms 586, 594
    - in reports 227, 245
    - inserting newline characters 227, 587
    - metacharacters 786
    - on toolbar buttons 25
    - platform differences 678
    - rotating in graphs 420
    - searching and replacing 34
    - setting color of 823

- text files
    - importing SQL statements from 130
    - line endings 677
    - saving 677
    - sharing 672
  - text patterns, matching in validation rules 296
  - text properties
    - in forms 587
    - in graphs 418
    - in reports 227
  - three-dimensional graphs
    - description 395
    - point of view 428
  - time
    - allowed for retrieval 61
    - checking string 772
    - converting to data type 850
    - DateTime data type 749
    - display formats for 273
    - minutes 797
    - now 803
    - relative 821
    - seconds 830, 831
  - Time function 850
  - Time keyword 274
  - timer, setting in reports 221
  - timestamps, used in updating rows 559
  - titles
    - in forms 582
    - in OLE server application windows 492
    - of graphs 391, 416
    - specifying text properties 418
  - Today function 851
  - Toolbar Item Command dialog box 29
  - toolbars
    - controlling display of 25
    - custom buttons 29
    - customizing 26
    - defining for executable file 619
    - description 24
    - docking 26
    - dropdown 24
    - in Form painter 565
    - in Report painter 198
    - moving 25
    - moving buttons 27
    - resetting 28
  - Toolbars dialog box 25
  - total of values
    - columns 708, 710, 847
    - crosstabs 741
    - running 745
  - Trail the Footer command 387
  - Trim function 852
  - TrueType fonts 678
  - Truncate function 853
  - truth table for expressions 698
  - TUTOR\_IM.PBL 47
- ## U
- Undo Clear command 580
  - Undo command
    - in Database Administration painter 97
    - in Report painter 204
  - Undo Size/Move command 580
  - unique indexes
    - creating 109
    - defining for primary key 105
  - unique keys, specifying for form 557
  - units of measure, specifying for reports 220
  - up and down arrows, in Quick Select 158
  - updatable columns in form 558
  - UPDATE statements
    - building in Database Administration painter 129
    - specifying WHERE clause 558
  - updates, in forms 542, 555
  - Upper function 854
  - uppercase 854
- ## V
- validation rules
    - about 92
    - applying to columns 294
    - customizing error messages 296
    - defining in Database painter 293
    - defining in Form painter 297
    - deleting 300
    - description 292

- expressions 706
- in forms 610
- maintaining 300
- newly entered value 761
- removing 295
- Value axis, graph 391
- values
  - checking for NULL 708, 767
  - detecting numeric 768
  - ensuring validity of 102
  - in graphs 391, 404
  - suppressing repeating 306
- Var function 855
- variables, SQLCache 383
- variance 855, 858
- VarP function 858
- View Definition dialog box 115
- View painter
  - description 70
  - opening 112
- views
  - creating 112
  - description 112
  - displaying in Select Tables dialog box 112
  - dropping 117
  - extended attributes of 93
  - opening 112
  - pipng 500
  - updating 555
- visible property, on the Expressions page 360

## W

- week, day of 751, 752
- WHERE clause
  - modifying when running forms 589
  - specified for update and delete 558
  - specifying in Quick Select 161
  - user modifying during execution 242
- width property, on the Expressions page 361
- Windows (Microsoft)
  - feature support 684
  - metafiles 672, 674, 684
- WMF files, adding to reports 246
- WMF image format 672, 674, 684

- WordCap function 861
- workspace
  - in Data Pipeline painter 504, 506
  - in Database painter 72
  - in Environment painter 48
  - in Form painter 565
  - in Report painter 195
- wrap height, default in freeform reports 154

## X

- x property, on the Expressions page 362
- x1, x2 property, on the Expressions page 362

## Y

- y property, on the Expressions page 363
- y1, y2 property, on the Expressions page 363
- Year function 862
- years in reports, specified with two digits 272

## Z

- zero display format 270
- zero, determining 832
- Zoom command
  - in print preview 211
  - in Report painter 204

