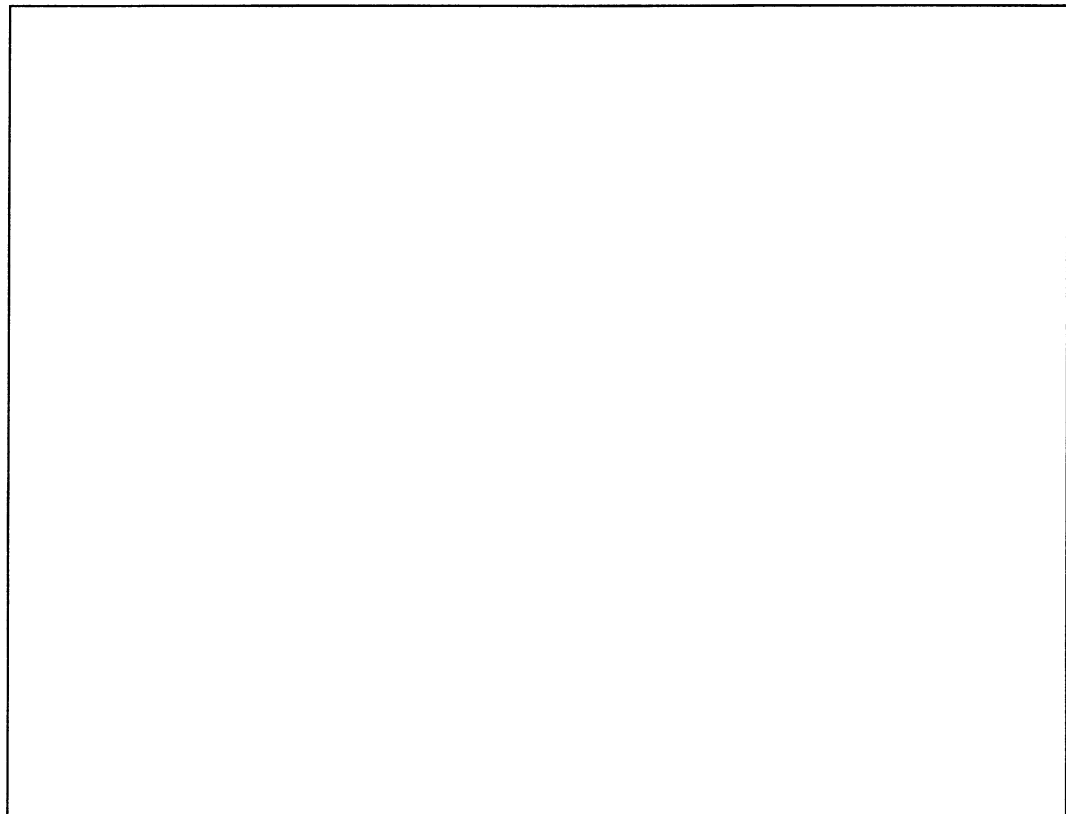


Introduction to GDL

Page Building with GDL

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY



What is GDL?

- ◆ Following slides show GDL basics
- ◆ GDL is Gentia's own programming language
- ◆ It is powerful and complex

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

This chapter will now talk about some of the basics of GDL. Naturally a topic as large as GDL cannot be covered in a course of this length. We touch on the basics here (and practise them) and in the following chapters we go into more detail and cover new areas as the application build requires.

Once the application is built, we will have looked at all the important and necessary elements of GDL and ensured a crucial platform from which to expand and grow.

Why Do We Need GDL?

- ◆ GDL is useful for:
 - quick fixes
 - complex manipulation
 - speed (sometimes)
- ◆ Maintenance of existing applications
- ◆ Enhance functionality

Alles was grafisch war, kann mit GDL

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

How Does GDL Work?

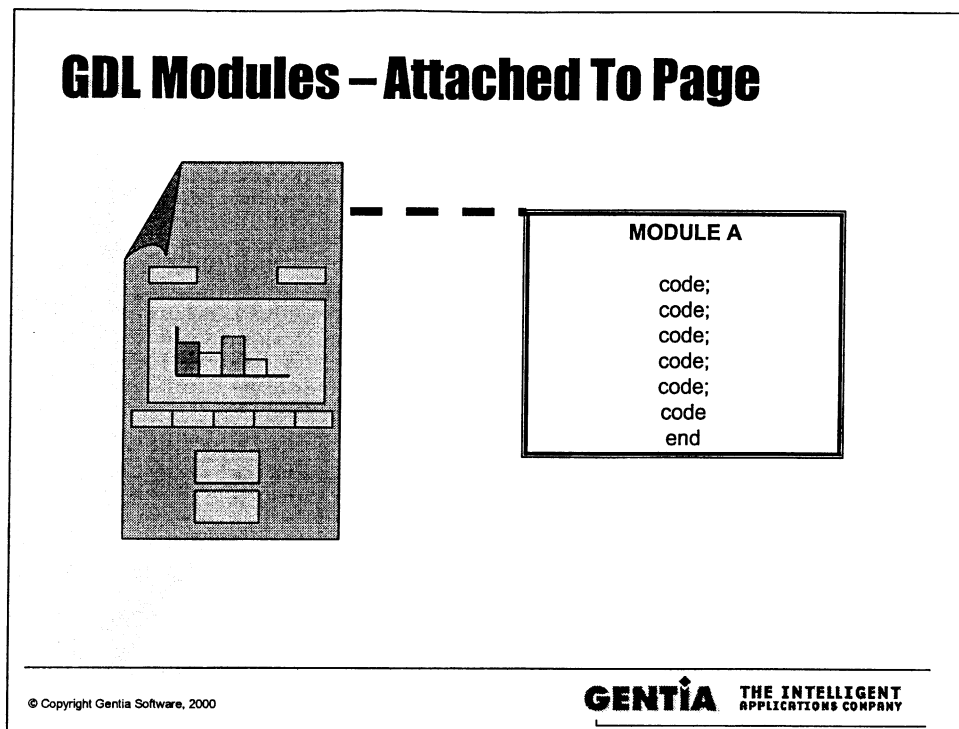
- ◆ GDL is held in modules
- ◆ Needs to be compiled
- ◆ Creates routines to store code
- ◆ Procedural

Case sensitive

statement optimization not ;"

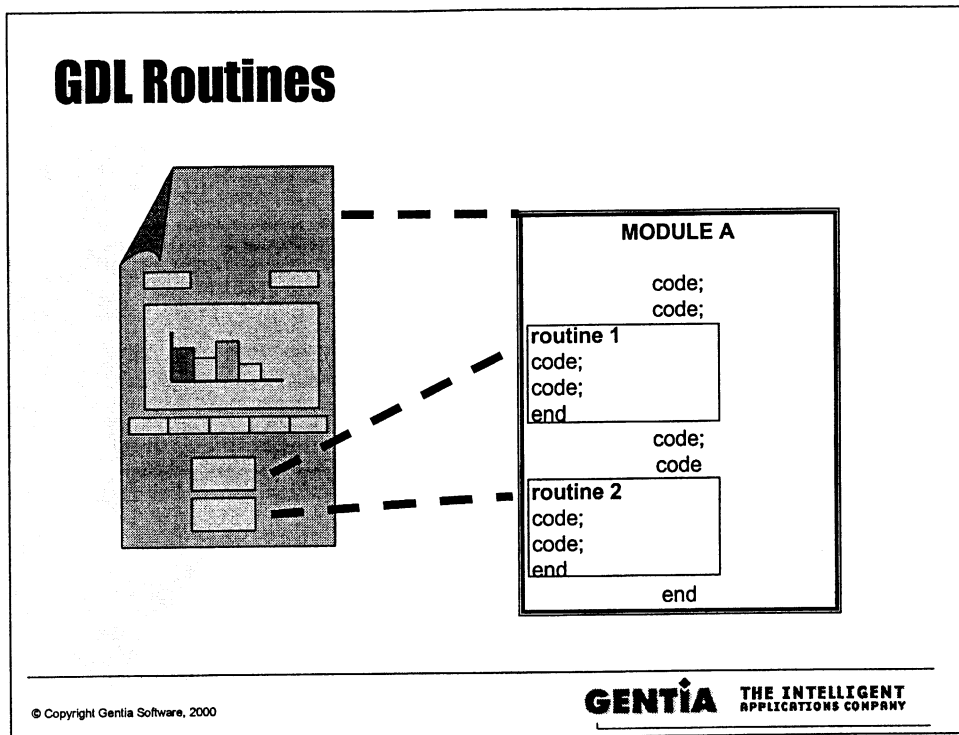
© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY



Each Gentia page can have one *GDL Module* attached to it. This GDL module will contain lines of compiled, sequential code. This code will be executed as soon as the page is accessed. It might typically contain actions specific to a page.

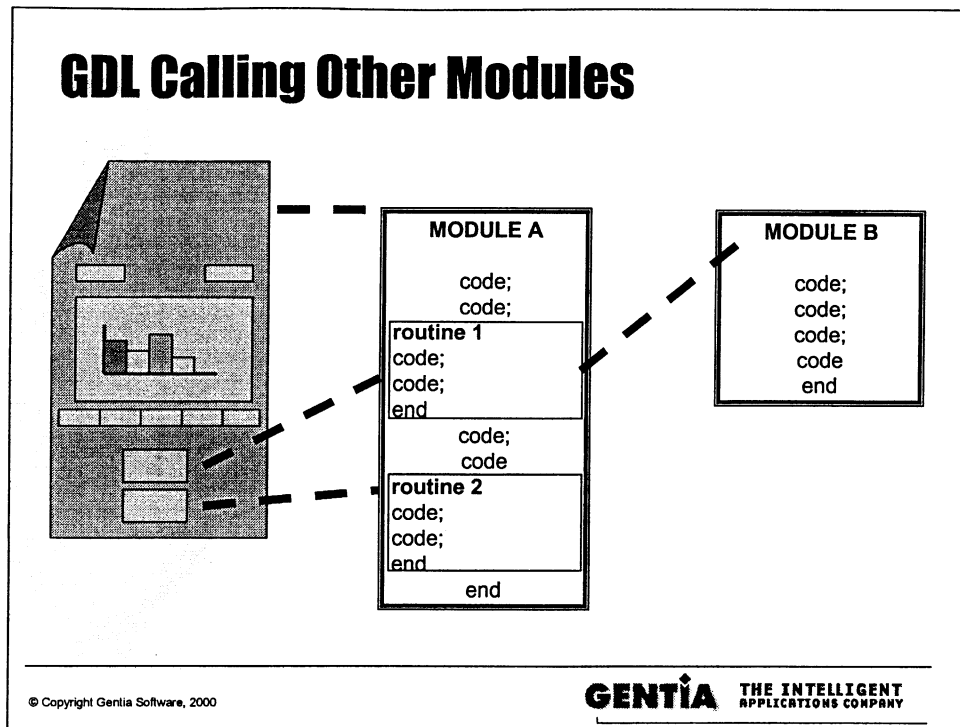
Occasionally, each module (or program) will need to be split into distinct chunks of code, which are executed at different times. Perhaps these sections of code might be called from a page widget. The sections are referred to as *routines*.



All code outside the routines will automatically be run when the page is first accessed, regardless of where it appears in the module. The code within a routine will be run when it is 'triggered' from elsewhere. In the above example, the routines are called from the two buttons.

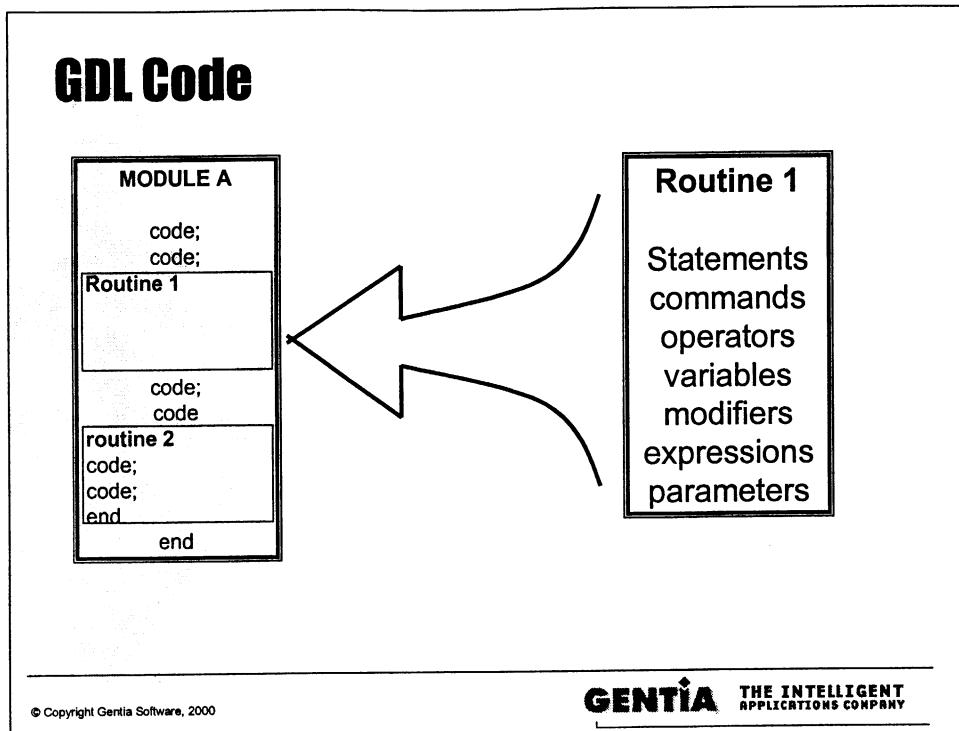
Each module can be regarded as a global routine, because it can be called from other modules or routines.

It is good coding practise to keep all module code together at the top of the module (above any routines) to aid maintainability.



The above example shows how **routine 1** in **Module A** is calling **Module B**. In this particular instance, module B's code is only executed from this one place. Of course, Module B might also be utilised from another module attached to another page, making it a global module.

A global module sits within an application but can be called at any time. We look at global modules in the next chapter.



GDL code is composed of a number of statements. Each statement is a complete instruction. The instructions are carried out sequentially, unless the code dictates otherwise. A GDL statement will contain some or all of the following elements:

- a **routine**, a subsection of code
- a **command**, providing the basic instruction for Gentia to execute
- **command operators**, qualifying or extending the command
- **variables**, containing information whose value may be changed either by the application code or the system system, during the execution of the application.
- **modifiers** are used to access the different types of information that Gentia maintains, with regards to connectors. They are also used to manipulate data flows and application code.
- **expression operators** qualify a command even further, by identifying a mathematical expression or a boolean operator which can be applied.
- **parameters** are usually a series of low-level statements which define variables.
- **end of statement** marker which is a semi-colon (;).

Example

LET \$wReportTab hidden = YES;

command	LET	The command to set a new value for a variable. The command is optional.
variable	\$wReportTab	Representing the name of a widget on the page.
modifier	hidden	The modifier that controls if a widget is displayed on a page or not.
expression	=	Qualifying the expression with a value (which follows)
valid value	YES	YES to hide; NO to show the widget
end of statement marker	;	

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Special Characters

- ◆ Some characters have special meaning

, ; “ #

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Some characters have special meaning within Gentia.

The , symbol is used in statements to separate parameter strings.

The ; symbol is used to indicate the end of a statement.

The “ symbol indicates a text string. The whole text string must be surrounded by double quotes.

The # symbol denotes a comment. All characters from this point to the end of the line are comments.

// →

|

Enkel
quote
NIET
in gelijke

Routines

- ◆ Start and End a routine
- ◆ ROUTINE.....IS.....END
- ◆ Code must be compiled

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Routines

A routine is a subsection of code, within a GDL module. A routine is started with the statement;

ROUTINE <name> **IS**

and is closed with the statement;

END

A routine is made up of a number of statements and each statement is closed by an *end of statement* marker (;) or an **END**.

Routine Example

```
ROUTINE <name1> IS
    xxxxxxxxx;
    xxxxxxxxx;
    xxxxxxxxx
END;      # end of routine1
```

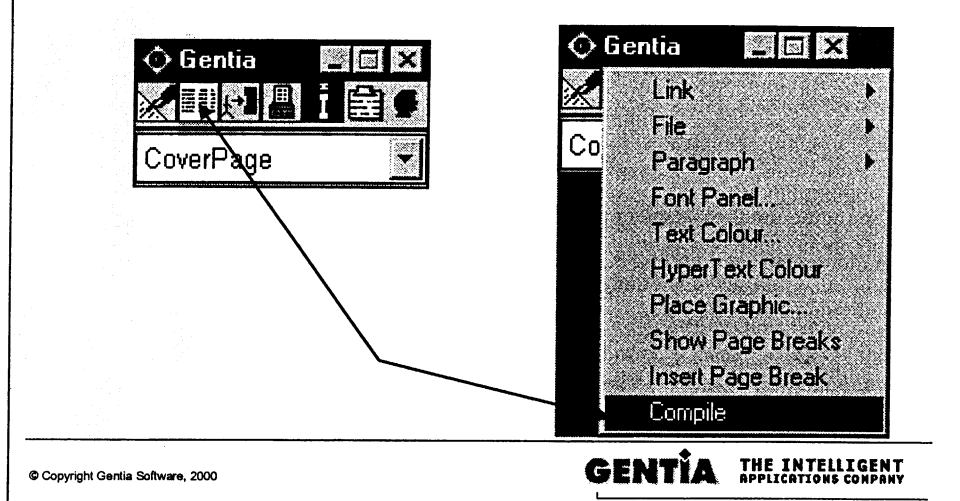
```
ROUTINE <name2> IS
    xxxxxxxxx;
    xxxxxxxxx
END      # end of routine2
END      # end of module
```

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Compiling

- ◆ All code must be compiled



Many programming languages are compiled. In simple terms, compiling the code will create an executable version of the language (as opposed to a text version) which will be stored internally within Gentic. It will run much faster than if it is not compiled.

With the GDL module selected, simply select the above buttons and the code will be compiled, with errors pointed out.

Use CTRL+Z to compile from the keyboard.

Executing a Routine

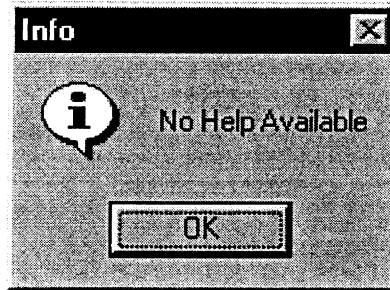
- ◆ User interaction
 - Button etc
- ◆ Calling a routine
 - Executes the routine from the code

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Calling a Routine

- ◆ CALL is used to initiate the execution of a routine
 - Parameters can be passed between routines
 - Use LET statements to define variables
 - **CALL** <routine name> (<parameter>)



© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Call

You use this command to initiate the execution of a routine, whether it is global to the application or local to the page. You may also pass parameters to be used in the routine by following the call with a series of LET statements which create new local variables for the routine about to be run. The parameters must be enclosed in parentheses and separated by commas.

It is good coding practise to prefix global routines with **!g**. Local routines have no prefix.

Example 1.

Routine *number1* calls the local routine *information1* which, in turn, uses a Gentia system function to display a message. Once this routine has completed its operations, control is passed back to the calling routine (*number1*), at which point any remaining commands are run.

```

MODULE: module1

  ROUTINE number1 IS
    CALL information1()
    .....
  END;

  ROUTINE information1 IS
    CALL !_infoAlert(&message = "Hello World")
  END;
  
```

Calling a Routine

- ◆ Routine *number2* calls the local routine *information2* passing a variable *¬e*. *Information2* will call the Gentia system function again, but this time, it will display the contents of the variable. Once this routine has completed its operations, control is passed back to the calling routine (*number2*), at which point any remaining commands are run.

```
MODULE: module2
  ROUTINE number2 IS
    CALL information2(&note = "Hello World")
    .....
  END;

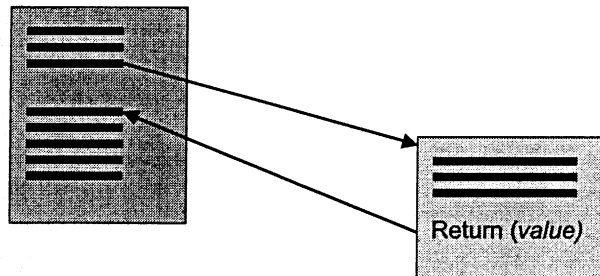
  ROUTINE information2 IS
    CALL !_infoAlert(&message = &note)
  END;
```

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Return

- ◆ Used to exit a routine AND return a value
- ◆ Use LET rather than CALL



© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

The return command is used to instruct Gentia to exit a routine and to return a value to the calling routine. The only rule is that the sub-routine is called using the LET command rather than the CALL command. It is also possible to use the command without parameters, perhaps at the end of an iterative statement. In this case, Gentia would return the value of the last statement processed. Alternatively, use **RETURN ()** or **RETURN 0**.

Return Example

```
ROUTINE main IS
  LET &Value1 = 10;
  LET &Value2 = 20;
  LET &result = GetValue (&Value1, &Value2)
END;
ROUTINE GetValue IS
  LET &newValue = (&Value1 + &Value2);
  RETURN (&newValue)
END
END
```

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Variables

- ◆ A storage location which contains a value
- ◆ The value may change during application
- ◆ One of five different types:
 - Boolean
 - Integer
 - Real
 - Text
 - Structure

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

A *variable* is a pre-defined area of memory which will be assigned a value from within the application. As the application is executed, the variable might change value. An example might be a counter to monitor how many times something happens. Consider the following example:

Een variabele die gebruikt wordt om de waarde
van een counter te bewaken.

Variable Example

```
◆ variable = 1;  
  Top;  
  Do something;  
  If variable = 5, goto end;  
  Add 1 to variable;  
  Goto Top;  
  End;
```

© Copyright Gentia Software, 2000

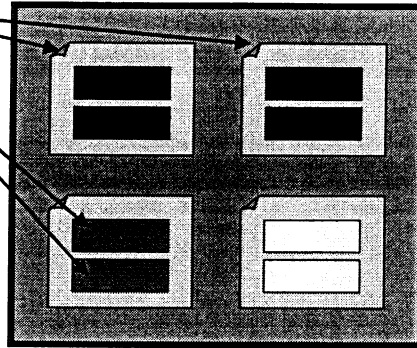
GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

- **Boolean** - is a value which can be either TRUE or FALSE;
- **Integer** - a number containing no decimal places
- **Real** - a number which can contain decimal places
- **Text** - containing text characters
- **Structure** - containing system structure variables, for example, names of widgets or fonts etc.

Variables

◆ Three different *availabilities* of variables

- Global
- Page
- Local



© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Global Variables

- ◆ Example which sets the reporting month for a series of application pages:

```
LET !reportingMonth = "September";
```



© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Global

A global variable name must start with a !, for example !var. A global variable exists from the moment it is created until it is either destroyed or the user exits Gentia. A global variable is used to carry values between Gentia pages, or when global routines are called. Typically, they are defined when the application is initiated and the first page is displayed. Consequently, global variables are often defined in a GDL module associated with CoverPage.

Page Variables

- ◆ Example which sets the month for which details are to be displayed in both a table widget and a chart widget on the same page, you would use the command:

```
LET $month = "September";
```



© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Page

A page variable name must start with a \$, for example \$var. A page variable exists from the moment the page is opened until it is either destroyed, or the page is exited. This type of variable is used to hold values that may be referenced from a number of different elements on the page.

Local Variables

- ◆ To set a counter which keeps track of the number of times an iterative statement is executed, you could use the statement:

```
LET &counter = 1;
```



© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Local

A local variable name must start with a **&**, for example **&var**. A local variable exists from the moment the routine containing it is started until it is either destroyed, or the routine is exited. However, it can be passed to another local routine.

N.B. Variable values are held in memory. In order to optimize your system memory, always ensure that the correct variable is used. Put simply, a global variable will use memory for the duration of the program and should only be used when absolutely necessary.

Naming variables is relatively straightforward. After choosing the relevant variable symbol (**!**, **\$** or **&**) the name can be any combination of letters, digits and underlines. The only rule is that the first character must be a letter. Variable names starting with underlines are reserved for system variables. Please refer to the Gentia Application Framework Standards Guide for Best Practice.

Structured Variables

- ◆ Entered manually
- ◆ Structured by Gentia

LET **&model** BE MODEL "Costmodel";

↑

call by reference

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Structured variables

Variable values can be manually entered, as in the previous examples, or automatically structured by Gentia. If a variable is created from a structured item, such as an EIS Model, Gentia will automatically build an index structure for it. For example:

LET **&model** BE MODEL "Costmodel";

This statement creates a structured variable **&model** which automatically contains an index structure based on the EIS Model definition of "Costmodel". This means the dimension names, coordinate codes and display strings can be accessed by reading the structured variable **&model**. For example:

LET **&DimensionName** be &model dimensions[1];

would set the variable **&DimensionName** equal to the first dimension in the EIS Model "Costmodel".

LET **&CoordinateName** be &model dimensions["Time"] coordinates[1];

would set the variable **&CoordinateName** equal to the first coordinate code in the dimension "Time" from the EIS Model "Costmodel".

Summary

- ◆ Understand user requirements
- ◆ Asking the right questions
 - Input Process Output
- ◆ Understanding basic GDL
 - More examples to follow!

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Practical 1 - 1 Display a Message

Aim: To create a GDL module that displays the value of a variable on a page.

	World	Africa	Nigeria	Abuja	South Africa	Johannesberg
Cost of Good Sold	248,430,395.19	12,599,166.43	3,292,030.66	3,292,030.66	4,841,037.24	4,841,037.24
Profit	42,389,410.92	2,247,761.37	641,326.96	641,326.96	833,392.34	833,392.34
Revenue	290,819,806.11	14,846,907.80	3,933,357.62	3,933,357.62	5,674,429.58	5,674,429.58
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.14	8	5,540.18	7,720.42	7,720.42

Run GDL routine one

Guidelines:

- Create a page with a table and a button using the Actual base model as the data source.
- You will need to connect a gdl module to the page.
- The button should run a gdl routine.
- The routine should create a variable and call the system function.
`!_infoAlert(&message = &variableName).`
- To compile the module use the text editor on the Gentia toolbar or keystroke "CTRL" + "Z".

Practical 1 - 2 Create an Array

Aim: To create a variable array and display its values.

	World	Africa	Nigeria	Abuja	South Africa	Johannesberg
Cost of Good Sold	248,430,395.19	12,599,156.43	3,292,030.66	3,292,030.66	4,841,037.24	4,841,037.24
Profit	42,389,410.92	2,247,751.37	641,326.96	641,326.96	833,392.34	833,392.34
Revenue	290,819,806.11	14,846,907.80	3,933,357.62	3,933,357.62	5,674,429.58	5,674,429.58
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.14	8	5,540.18	7,720.42	7,720.42

Buttons: Run GDL routine one, Run GDL routine two

Guidelines:

- Use the previously created page and GDL module.
- The array should contain the following values:

Array	Value
&locate[1]	wor00
&locate[2]	afr01
&locate[3]	asi01
&locate[4]	eur01
&locate[5]	nam01
&locate[6]	sop01

- !_infoAlert windows are used to display the array.

Conditions and Loops

Conditions

A conditional statement always contains an expression that can be evaluated as either true or false. This may be a numeric comparison or a string comparison, e.g. to check a user name and password.

When creating conditional statements, Gentia provides a number of commands and associated operators that may be used to check the condition. Each condition can then be followed by a series of statements to be executed when the condition is met. Where required, you may nest conditional statements.

To generate conditional statements, the IF ... THEN ... FI construct is used, whilst there are several variations, as detailed below:

IF....THEN.....FI

This is a simple conditional statement. IF initiates the condition, and FI indicates the end of the condition statement.

Example

```
IF &a = 5 THEN &b="YES" FI;
```

IF....THEN.....FI with BEGIN and END

If you want a number of statements to be executed when the condition is true, you must add the BEGIN and END operators.

Example

```
IF &a = 1 THEN
  BEGIN
    LET &a=0;
    LET &c=30;
    INCREMENT &b BY 10
  END
FI;
```

IF....THEN.....FI with ELSE

When you want one series of statements to be executed if a condition is true and another if the condition is not, you must add an ELSE operator.

Example

```
IF &a = 1 THEN
  BEGIN
    LET &a=0;
    LET &c=30
  END
ELSE
  INCREMENT &b
FI;
```

Other operators that may be used with IF...THEN....FI are AND, OR and NOT.

Examples

```
IF &a=&b AND &c>=&b THEN.....FI;  
IF &a=&b OR &a=&c THEN ..... FI;  
IF NOT &a=&b THEN .....FI;
```

Loops

A series of instructions performed repeatedly until some specified condition is satisfied

FORDO

The FOR....DO construct allows you to create loops that evaluate an expression first and only continue to execute if the result of that expression is true.

Example

```
ROUTINE fordo is  
  LET &c=1;  
  FOR DO  
    BEGIN  
      IF &c>3 THEN BREAK FI;  
      CALL !_infoAlert(&message="Loop again");  
      INCREMENT &c  
    END  
  END  
END
```

FOR ...NEXT.....DO

You use this construct to define a statement to be executed every time the system passes through the loop.

Example 1

```
FOR &i = 1 NEXT INCREMENT &i DO .....
```

Example 2

```
ROUTINE fornextdo is  
  FOR &i=1 NEXT INCREMENT &i DO  
    BEGIN  
      IF &i >3 THEN BREAK FI;  
      CALL !_infoAlert(&message="Loop again")  
    END  
  END  
END
```

FOR EACH.....IN....DO

You use this construct to perform actions for each item in a variable index.

Example 1

```
FOR EACH INTEGER &a IN $array DO.....
```

where \$array is an indexed variable and the integer index values are successively read into &a before executing the subsequent statements.

The condition ends when all index values of the selected type have been read.

Example 2

```
LET $exrate[1]="US Dollar";  
LET $exrate[2]="French Franc";  
LET $exrate[3]="Japanese Yen";
```

ROUTINE arrayvariable IS

```
    FOR EACH INTEGER &c IN $exrate DO  
        CALL !_infoAlert(&message=$exrate[&c])  
    END  
END
```

Example 3

```
LET $exrate["USD"]="US Dollar (USD)";  
LET $exrate["FF"]="French Franc (FF)";  
LET $exrate["YEN"]="Japanese Yen (YEN)";
```

ROUTINE arraytext

```
    FOR EACH TEXT &c IN $exrate DO  
        CALL !_infoAlert(&message=$exrate[&c])  
    END  
END
```

FOR....WHILE.....DO

You use this construct to check a condition before performing the actions specified for the loop, and to break out of the loop if the condition is not met.

Example 1

```
FOR &c=5 WHILE &a <10 DO .....
```

This example will execute the specified statements for &a values in the range 1-9.

Example 2

```
FOR &c=5 DO..... WHILE &a<10 ;
```

This example will execute the specified statements for &a values in the range 1-10.

FOR.....UNTIL.....DO

You use this construct to execute a statement and then evaluate an expression, and to break out of the loop if the specified condition is not met.

Example

```
FOR &c=5 UNTIL &a=10 DO;
```

This example will execute the specified statements for &a in the range 1-10.

Practical 1 - 3 Use a Loop

Aim: Use a loop to step through an array and display it on !_infoAlert windows.

The screenshot shows a software application window titled "Partner Training Accreditation - AppDev Three". The window contains a table with financial data and an "Info" dialog box.

	World	Africa	Nigeria	Abuja	South Africa	Johannesberg
Cost of Good Sold	248,430,395.19	12,599,156.43	3,292,030.66	3,292,030.66	4,841,037.24	4,841,037.24
Profit	42,389,410.92	2,247,751.37	641,326.96	641,326.96	833,392.34	833,392.34
Revenue	290,819,806.11	14,846,907.80	3,933,357.62	3,933,357.62	5,674,429.58	6,674,429.58
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.		5,540.18	7,720.42	7,720.42

An "Info" dialog box is overlaid on the table, displaying the text "Coordinate name: not00" and an "OK" button.

Below the table, there are three buttons labeled "Run GDL routine one", "Run GDL routine two", and "Run GDL routine three".

Guidelines:

- Create the array for reuse when the page opens with code outside of the routine statements using a page variable.

Array	Value
\$locate[1]	world
\$locate[2]	Europe
\$locate[3]	NthAm
\$locate[4]	SthPac
\$locate[5]	Asia
\$locate[6]	Africa

- Create a loop to step through the array. The pseudo-code below shows a conditional loop that will perform a statement for every member of the array.
- Display the coordinate alongside a text string eg. "Coordinate name is" + \$locate[&i]

Pseudo-code

FOR EACH INTEGER &a IN \$locate DO...

Modifiers

Modifiers are a fundamental aspect of GDL. They identify a particular piece of information associated with a widget or connector, which can be modified.

A modifier in a statement is rather like the verb in a sentence, as it specifies what is to be done to the widget or connector, i.e. how it is to be modified.

All widgets and connectors on a page can automatically be recognised as page variables, with structures. Each attribute, (or piece of information) associated with the widget or connector forms a different element of the variable. Modifiers are used to identify each of these attributes.

Most of the modifier names are easily recognisable if you are familiar with widget operations, and using them appropriately will allow you to identify and/or change individual attributes.

For example, the *contents* modifier can be used to control the contents of an edit box called:

```
LET $wEdtOne contents = "The modifier is contents".
```

Modifiers are case sensitive and tend to be object related, as shown in the following examples. The modifiers appear in *italics*:

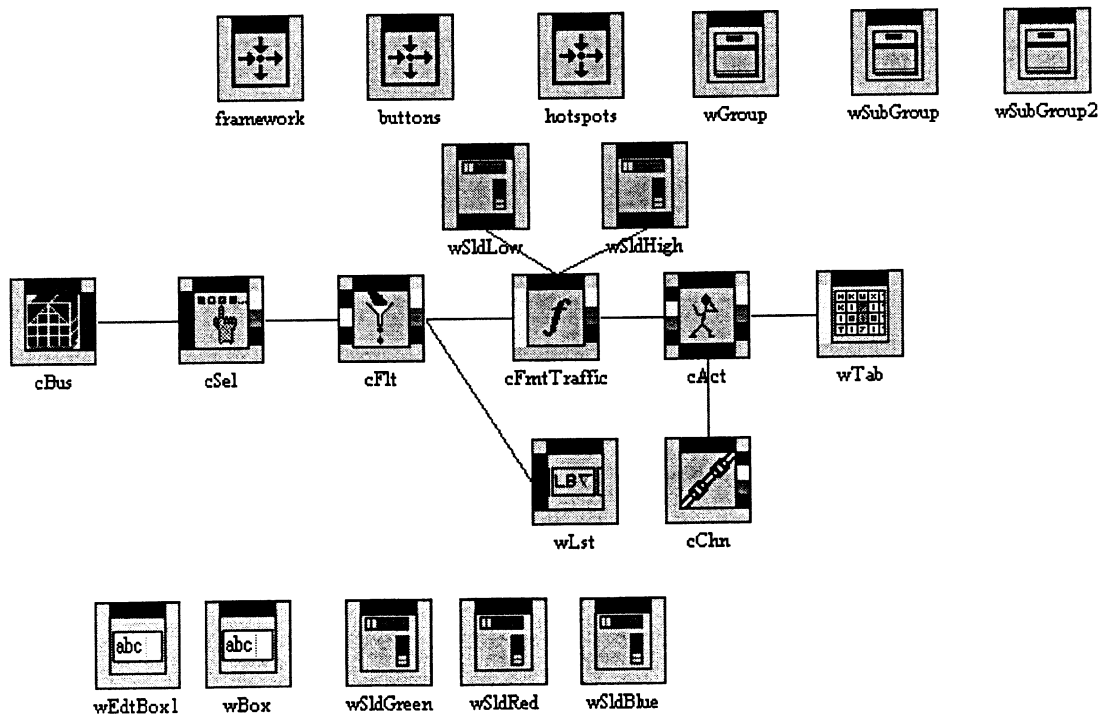
```
LET $wTabOne hidden = YES           hides a tables  
LET $wChtOne hidden = NO           reveals a chart  
LET &localVariable = $wLstOne selectedEntry  assigns listbar selection to variable
```

Example

Aim: To demonstrate the power of modifiers.

	Cost of Good Sold	Profit	Revenue	Unit Cost
Abuja		641,326.96	3,933,357.62	N/A
Johannesberg	4841037.24	833,392.34	5,674,429.58	N/A
Harare	4466088.53	773,032.08	5,239,120.60	N/A
Montreal		0.00	0.00	N/A
Toronto		1,123,295.52	7,648,433.01	N/A
Mexico City	3653944.89	635,701.56	4,289,646.46	N/A
Acapulco		0.00	0.00	N/A
San Francisco		1,897,361.65	12,477,096.31	N/A
Chicago		1,151,225.37	7,793,688.59	N/A
New York		1,166,238.24	12,567,578.55	N/A
Sydney		1,080,721.92	7,316,451.08	N/A
Auckland		0.00	0.00	N/A
Tokyo		4,035,271.73	27,093,864.58	N/A
Kuta	4716407.81	820,207.67	5,536,615.48	N/A
Antwerp		2,000,000.00		
Bruges	4731410.20	81		

Connections Mapper:



This page allows you to select coordinates, and enter numbers into the slider bar edit fields. The result will be to create a format on those coordinates with the selected colours, based on the required limits. Note the modifiers that have been used below.

```
LET $wBox background = COLOUR(1,178,1);
LET $red = 1;
LET $green =178;
LET $blue = 1;
LET $flag = 0;
```

```
LET $xCoord = 410;
LET $yCoord = 390;
```

```
ROUTINE color IS
    LET $green=integer($wSldGreen currentValue);
    LET $blue=integer($wSldBlue currentValue);
    LET $red=integer($wSldRed currentValue);
    LET $wBox background = colour($red,$green,$blue)
END;
```

```
ROUTINE one IS
```

```
LET $cFmtTraffic currentFormat = 1;
LET $cFmtTraffic currentSlice = 1;
LET $cFmtTraffic lowLimitsBackground = colour($red,$green,$blue)
```

```
END;
```

```
ROUTINE two IS
```

```
LET $cFmtTraffic currentFormat = 1;
LET $cFmtTraffic currentSlice = 1;
LET $cFmtTraffic midLimitsBackground = colour($red,$green,$blue)
```

```
END;
```

```
ROUTINE three IS
```

```
let $cFmtTraffic currentFormat = 1;
let $cFmtTraffic currentSlice = 1;
let $cFmtTraffic highLimitsBackground = colour($red,$green,$blue)
```

```
END;
```

```
ROUTINE hide IS
```

```
LET $wGroup hidden = NO;
LET $wBut4 hidden = YES
```

```
END;
```

```
ROUTINE unhide IS
```

```
LET $wGroup hidden = YES;
LET $wBut4 hidden = NO
```

```
END;
```

ROUTINE toTheBottom IS

```
LET $yCoord = $yCoord + 15;  
LET $wGroup xOrigin = $xCoord;  
LET $wGroup yOrigin = $yCoord
```

END;

ROUTINE toTheTop IS

```
LET $yCoord = $yCoord - 15;  
LET $wGroup xOrigin = $xCoord;  
LET $wGroup yOrigin = $yCoord
```

END;

ROUTINE toTheRight IS

```
LET $xCoord = $xCoord + 15;  
LET $wGroup xOrigin = $xCoord;  
LET $wGroup yOrigin = $yCoord
```

END;

ROUTINE toTheLeft IS

```
LET $xCoord = $xCoord - 15;  
LET $wGroup xOrigin = $xCoord;  
LET $wGroup yOrigin = $yCoord
```

END;

ROUTINE formatCoord IS

```
IF $flag = 0 THEN  
  BEGIN  
    LET $cFmtTraffic addCoordinate = $wEdtBox1 contents;  
    LET $flag = 1  
  END  
ELSE  
  BEGIN  
    LET $cFmtTraffic addFormat = YES;  
    LET $cFmtTraffic addCoordinate = $wEdtBox1 contents;  
    LET &low = integer(20);  
    LET &high = integer(1000);  
    LET $cFmtTraffic lowLimitsAre = &low;  
    LET $cFmtTraffic highLimitsAre = &high  
  END  
FI  
  
END  
END
```

Practical 1 - 4 Create an Application

Aim: To use an array and looping procedures to create a small application.

The screenshot shows a web browser window titled "Partner Training Accreditation - AppDev Four". The browser's address bar and menu items are visible. The main content area displays a table with financial data across seven regions: World, Africa, Asia, Europe, North America, and South Pacific. Below the table are four buttons labeled "Run GDL routine one" through "Run GDL routine four".

	World	Africa	Asia	Europe	North America	South Pacific
Cost of Good Sold	248,430,395.19	12,599,156.43	27,775,000.66	163,017,888.36	38,802,620.58	6,235,729.16
Profit	42,389,410.92	2,247,751.37	4,855,479.39	28,231,635.89	5,973,822.34	1,080,721.92
Revenue	290,819,806.11	14,846,907.80	32,630,480.06	191,249,524.25	44,776,442.92	7,316,451.08
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.14	27,267.34	175,232.88	63,463.18	9,872.28

Buttons below the table:

- Run GDL routine one
- Run GDL routine two
- Run GDL routine three
- Run GDL routine four

Guidelines:

- Add a filter to the connections mapper called "cFit" and make it positive.
- Add "Location" as the dimension in the filter as the page opens.
- Use a loop to run through the array coordinates.
- Use a conditional statement in conjunction with the filters' modifiers to first makeFilter and then addToFilter to display only the array coordinates on the table.

Pseudo-Code

```

IF first time THEN
    makeFilter
ELSE
    addToFilter
  
```

Practical 1 - 5 Optional Exercise

Aim: To make the application look more professional.

	World	Africa	Asia	Europe	North America	South Pacific
Cost of Good Sold	248,430,395.19	12,599,166.43	27,775,000.66	163,017,888.36	38,802,620.58	6,235,729.16
Profit	42,389,410.92	2,247,751.37	4,955,479.39	28,231,635.89	5,973,822.34	1,080,721.92
Revenue	290,819,806.11	14,846,907.80	32,630,480.06	191,249,524.25	44,776,442.92	7,316,451.08
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.14	27,267.34	175,232.88	63,463.18	9,872.28

Run GDL routine one Run GDL routine two Run GDL routine three Run GDL routine four Run GDL routine five

Guidelines:

- Use the modifier holdOff in conjunction with the filter
- For example: LET \$cFit holdOff = YES
- This has the effect of stopping any changes to the filter from being displayed until holdOff is set to NO.

Introduction to GDL

2 Structured Variables

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Structured Variables

Structured variables can be used to hold information about the structure of a specified model. If this information is required in more than one routine within a module, then the structured variable should be defined as a page variable outside the routine structure to aid maintainability. For example,

```
$model BE MODEL "Salemodel"; - where Salemodel is the model spec name.
```

It is possible to read a structured variable containing a model structure and return the co-ordinate code and description. The keyword **name** returns the co-ordinate code and the keyword **displayAs** returns the co-ordinate description. For example:

```
$model BE MODEL "Salemodel";  
FOR EACH INTEGER &i IN $model dimensions["Time_Dimension"] coordinates DO  
BEGIN  
$monthcode[&i]=$model dimensions["Time_Dimension"] coordinates[&i] name;  
$monthdesc[&i]=$model dimensions["Time_Dimension"] coordinates[&i] displayAs  
END
```

This will create two page variable arrays, each with integer indices or pointers, where \$monthcode contains the co-ordinate codes for the Time dimension and \$monthdesc contains the co-ordinate descriptions for the Time dimension.

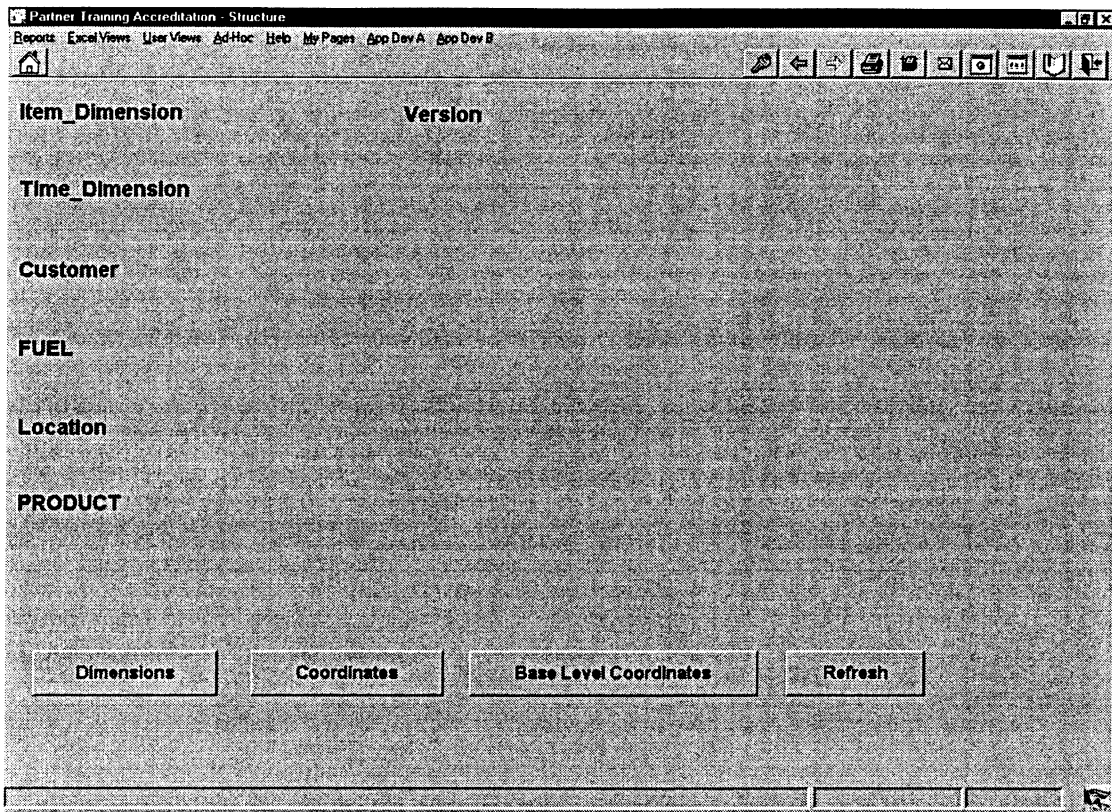
The above example could be made more efficient using the BE command, for example:

```
$model BE MODEL "Salemodel";  
$modeltime be $model dimensions["Time_Dimension"];  
FOR EACH INTEGER &i IN $modeltime coordinates DO  
BEGIN  
$monthcode[&i]=$modeltime coordinates[&i] name;  
$monthdesc[&i]=$modeltime coordinates[&i] displayAs;  
END
```

Given a co-ordinate description it is not possible to directly read the co-ordinate code using a structured variable.

Example

Aim: This example steps through the dimensions of a business model.



```
# -----  
# Module:      PTA_Mod_Structure  
#  
# Comments:   This module is an example of reading a model structure  
# -----
```

```
LET $model BE MODEL "Actual";
```

```
$box[1] be $wEdt1;  
$box[2] be $wEdt2;  
$box[3] be $wEdt3;  
$box[4] be $wEdt4;  
$box[5] be $wEdt5;  
$box[6] be $wEdt6;  
$box[7] be $wEdt7;  
$box[8] be $wEdt8;  
$box[9] be $wEdt9;  
$box[10] be $wEdt10;  
$box[11] be $wEdt11;  
$box[12] be $wEdt12;  
$box[13] be $wEdt13;  
$box[14] be $wEdt14;  
$box[15] be $wEdt15;  
$box[16] be $wEdt16;  
$box[17] be $wEdt17;
```

↳ gelijk aan naam op scherm

```
# -----  
# Read model dimensions  
# -----
```

```
ROUTINE one IS
```

```
FOR EACH INTEGER &i IN $model dimensions DO  
  BEGIN  
    LET $box[&i] contents = $model dimensions[&i] name;  
    CALL !_sleep(&seconds=1)  
  END  
END;
```

Practical 2 - 1 Read Dimensions

Aim: Create a GDL module that reads and displays all the dimensions in the model

	World	Africa	Nigeria	Abuja	South Africa	Johannesberg
Cost of Good Sold	248,430,395.19	12,599,156.43	3,292,030.66	3,292,030.66	4,841,037.24	4,841,037.24
Profit	42,389,410.92	2,247,751.37	641,326.96	641,326.96	833,392.34	833,392.34
Revenue	290,819,806.11	14,846,907.80	3,933,357.62	3,933,357.62	5,674,429.58	5,674,429.58
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A		N/A	N/A	N/A
Units Sold	296,048.81	20,213.14		5,540.18	7,720.42	7,720.42

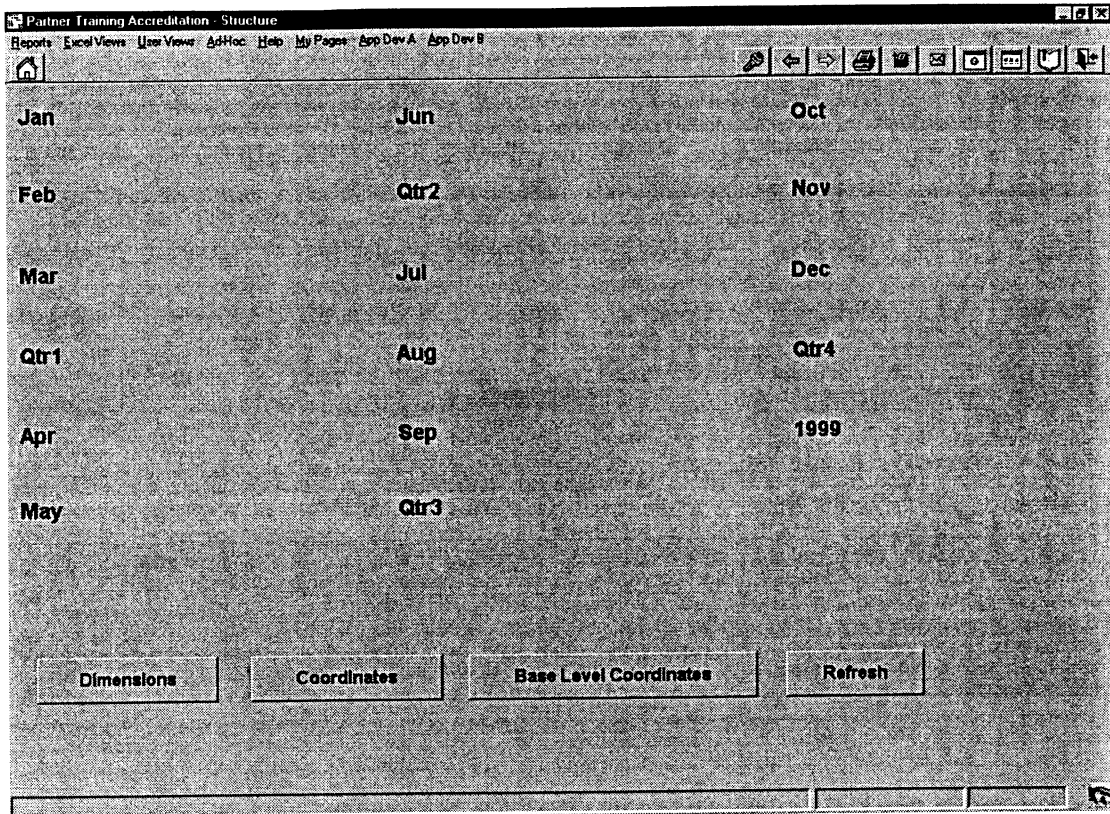
Run GDL routine six

Guidelines

- Copy or create a new page still using the Actual base model as the source data.
- Create a new module and routine.
- Create a variable that has the same structure as the Actual base model.
- Read through the model variable by integer.
- Use the modifiers "dimensions" and "name".
- Use !_infoAlert to display the model's dimensions.

Example

Aim: To display all the co-ordinate display strings in the Time dimension.



```
# -----
# Read co-ordinates in the Time dimension using the co-ordinate numbers
# -----
```

```
ROUTINE two IS
```

```
FOR EACH INTEGER &i IN $model dimensions["Time_Dimension"] coordinates DO
    BEGIN
        LET $box[&i] contents = $model dimensions["Time_Dimension"]
            coordinates[&i] displayAs;
        CALL !_sleep(&seconds=1)
    END
END;
```

Practical 2 - 2 Read Coordinates

Aim: Read and display all the co-ordinates in the Location dimension.

	World	Africa	Nigeria	Abuja	South Africa	Johannesberg
Cost of Good Sold	248,430,395.19	12,599,156.43	3,292,030.66	3,292,030.66	4,841,037.24	4,841,037.24
Profit	42,389,410.92	2,247,751.37	641,326.96	641,326.96	833,392.34	833,392.34
Revenue	290,819,806.11	14,846,907.80	3,933,357.62	3,933,357.62	5,674,429.58	5,674,429.58
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.14	5,540.18	5,540.18	7,720.42	7,720.42

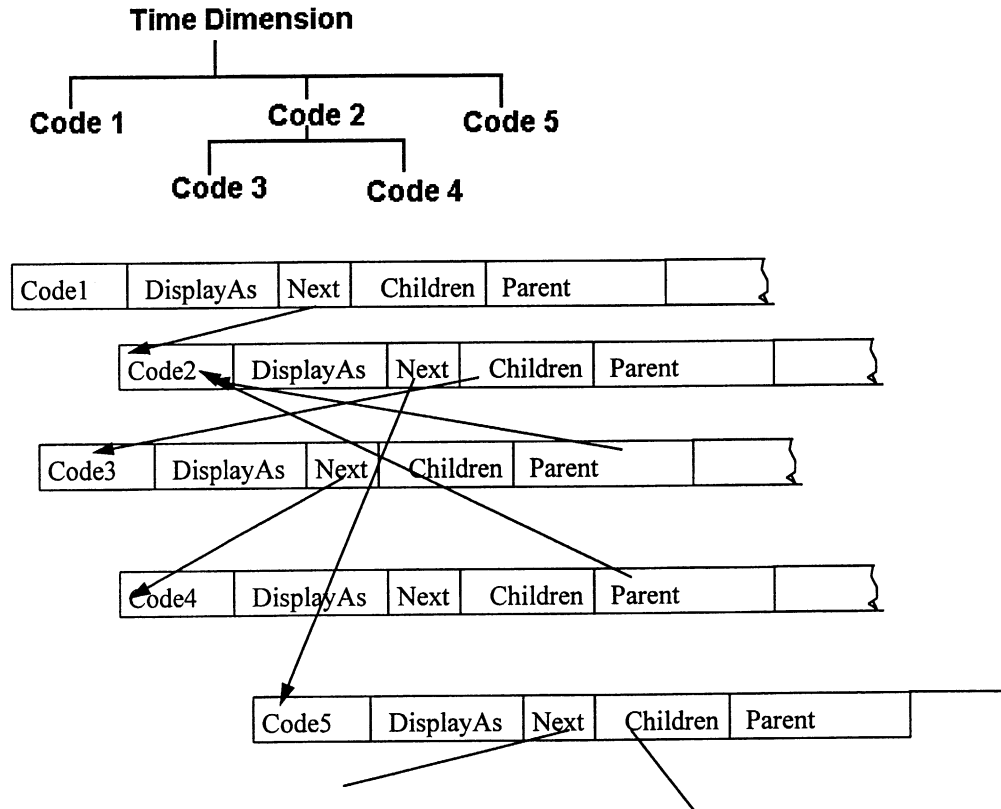
Run GDL routine seven

Guidelines

- Create a variable that has the same structure as the Location dimension.
- Read through the model variable by integer.
- Use the modifiers "dimensions", "coordinates" and "displayAs".
- Use !_infoAlert to display the co-ordinates in the Location dimension.

Base Members

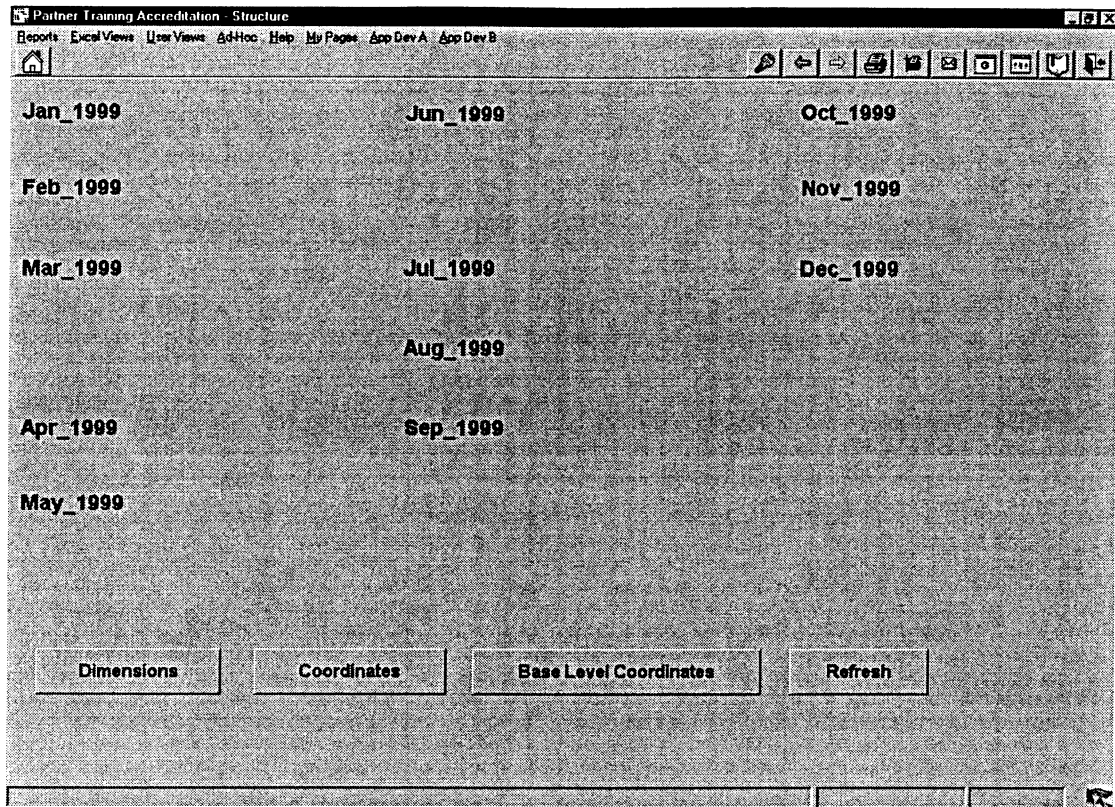
Below is a diagram of how conceptually the data dictionary is stored within Gentia.



Next, **Children**, and **Parent** are pointers. The Children pointer returns the number of children for a given code. The Parent pointer returns the Parent code for a given code. Once you have the number of children for a given code a loop may be written to move to each child in turn for the given code.

Example

Aim: To display all the base level co-ordinate names in the Time dimension.



```
# -----
# Read co-ordinates of the Time dimension using the
# co-ordinate numbers and display the base level co-ordinates
# -----
```

```
ROUTINE three IS
```

```
FOR EACH INTEGER &i IN $model dimensions["Time_Dimension"] coordinates DO
  BEGIN
    LET &coord BE $model dimensions["Time_Dimension"] coordinates[&i];
    IF &coord children = 0 THEN
      BEGIN
        LET $box[&i] contents = &coord name;
        CALL !_sleep(&seconds=1)
      END
    FI
  END
END;
END;
```

Practical 2 - 3 Base Level Coordinates

Aim: To read and display all of the base level co-ordinates within the Location dimension.

	World	Africa	Nigeria	Abuja	South Africa	Johannesberg
Cost of Good Sold	248,430,395.19	12,599,166.43	3,292,030.66	3,292,030.66	4,841,037.24	4,841,037.24
Profit	42,389,410.92	2,247,751.37	641,326.96	641,326.96	833,392.34	833,392.34
Revenue	290,819,806.11	14,846,907.80	3,933,357.62	3,933,357.62	5,674,429.58	5,674,429.58
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	296,048.81	20,213.14	8	5,540.18	7,720.42	7,720.42

Run GDL routine eight

Guidelines

- Create a variable that has the same structure as the Location dimension.
- Read through the model variable by integer.
- Use the modifiers "dimensions", "coordinates" and "name".
- Check each co-ordinate for children using the modifier "children".
- If children = 0 then use !_infoAlert to display the coordinate name.

Practical 2 - 4 Create an Application

Aim: To apply what we have learnt to the table.

	Abuja	Johannesberg	Harare	Montreal	Toronto	Mexico City
Cost of Good Sold	3,292,030.66	4,841,037.24	4,466,088.53	0.00	6,525,137.50	3,653,944.89
Profit	641,326.96	833,392.34	773,032.08	0.00	1,123,295.52	636,701.56
Revenue	3,933,357.62	5,674,429.58	5,239,120.60	0.00	7,648,433.01	4,289,646.46
Unit Cost	N/A	N/A	N/A	N/A	N/A	N/A
Unit Price	N/A	N/A	N/A	N/A	N/A	N/A
Units Sold	5,540.18	7,720.42	6,952.54	0.00	8,065.58	5,712.77

Run GDL routine nine

Guidelines

- Use "addDimension" to add the Location dimension when the page opens.
- Create a variable that has the same structure as the Location dimension.
- Read through the model variable by integer.
- Use the modifiers "dimensions", "coordinates", and "name".
- Create a variable of a coordinate.
- Use the specified variable to check for children.
- If no children then first use makeFilter and then addToFilter to create a filter just showing the base level co-ordinates of the Location dimension on the table.

Practical 2 - 5 Optional Exercise

Aim: To restrict the table to show only base level co-ordinates in all dimensions.

The screenshot shows a software application window titled "Partner Training Accreditation - AppDev Ten". The window contains a data table with the following structure:

	Jan	Feb	Mar	Apr	May	Jun
fleet	12,790.80	12,790.80	12,790.80	12,790.80	12,790.80	12,790.80
business	29,551.86	29,551.86	29,551.86	29,551.86	29,551.86	29,551.86
private	206,863.05	88,655.59	59,103.73	14,184.89	103,431.53	124,117.83

Below the table, there are several dropdown menus and a button:

- Cost of Good Sold
- Petrol
- Class One
- Actual
- Abuja
- Run GDL routine ten

Guidelines:

- For example: LET \$cFlt holdOff = YES
- Create a routine to loop through the dimensions in model.
- Use "CALL" to run another routine.
- Within the call statement send the current dimension name.
- Use the "children" modifier to discover the base level co-ordinates of the dimension.
- Make a filter of the base level co-ordinates of the current dimension
- LET \$cFlt holdOff = NO.
- Add slice, row and column listbars to the page.
- Emphasise the functionality by giving the breakable and backtrack headers different colours.

Introduction to GDL

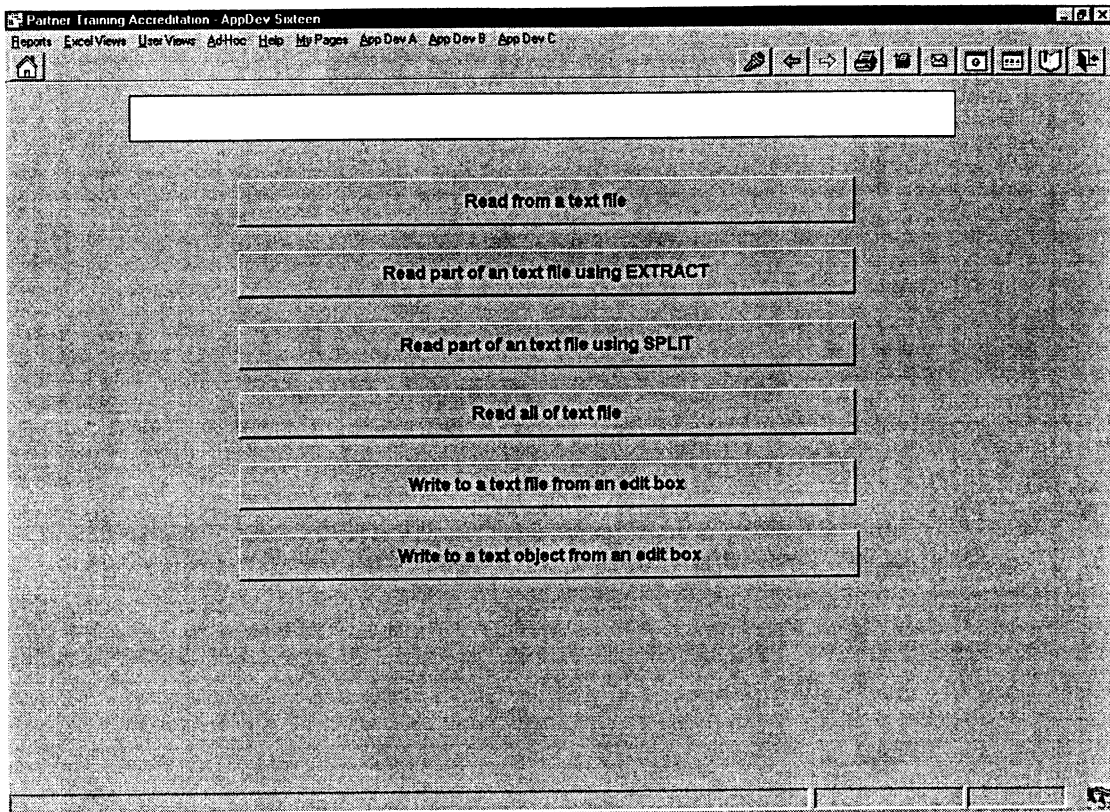
3 File Handling

© Copyright Gentia Software, 2000

GENTIA THE INTELLIGENT
APPLICATIONS COMPANY

Overview of Exercises

Aim: To understand how to manipulate text and data files.



The practical is split up into six related areas:

- Read a line from an external file;
- Read part of the file using EXTRACT;
- Read part of the file using SPLIT;
- Read all the lines from an external file;
- Write to an external file;
- Write to a text object

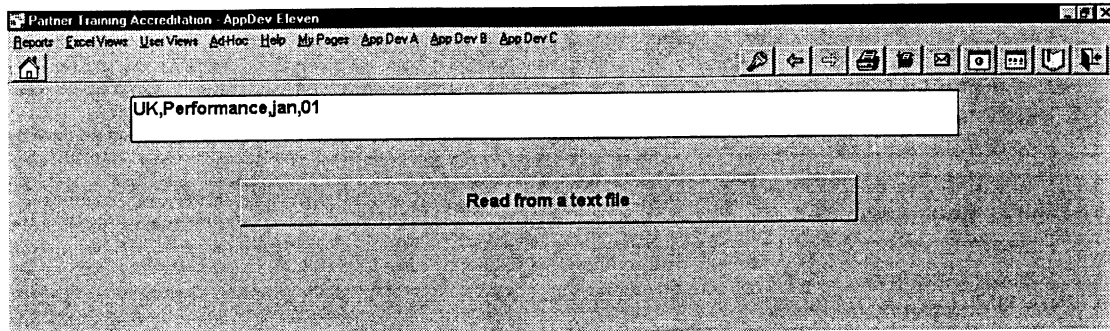
The data is shown below in a comma separated value CSV format. The first field is location, the second version, the third time, and the fourth holds the numeric value of a member from the item dimension.

The data file :

UK,Performance,jan,01
UK,Performance,feb,02
UK,Performance,mar,04
UK,Performance,apr,08
UK,Performance,may,16
UK,Performance,jun,32
UK,Performance,jul,64
UK,Performance,aug,128
UK,Performance,sep,256
UK,Performance,oct,512
UK,Performance,nov,1024
UK,Performance,dec,2048

Practical 3 - 1 Read a Line

Aim: To read one line from an external file.



Guidelines

- Read the external file into a variable.
- Read one line into another variable.
- Display the line variable in the edit box.

To help you do this there are more Gentia reserved words you need to use, "BE FILE READONLY" and the modifier "nextLine". Also when defining a files path, Gentia needs to see "\\" rather than "\".

Pseudo Code

ROUTINE Read IS

```
LET Variable1 BE READONLY FILE pathname;  
LET Variable2 = Variable1 nextLine;  
LET EditBox contents hold Variable2
```

END;

Extract

Once you have the current line of the text file held within a variable you can perform a number of actions upon it. Separating fields from a line of data can be done in two ways, by using EXTRACT or SPLIT.

This example uses the EXTRACT function.

The EXTRACT function is used to separate a text string into a number of sub-strings. The extraction is controlled by the start position of a character on the original string, and the length of the string.

Three parameters are required:

- The variable to be split
- The start position
- The length of the string.

```
LET <variable1> = EXTRACT(<variable2>, "<start position for extract>", <no. of characters to be extracted>);
```

This extracts from *variable2* a string on every start position and for a set length and stores it in *variable1*.

Note that the first sub-string in a string starts with position 0.

Examples

```
LET &Ingredient1 = EXTRACT(&recipe,0,10)
```

if &recipe was

Chick Peas Olive Oil Tahini Lemon Juice Yoghurt Garlic

after using EXTRACT the variable &Ingredient1 would contain:

Chick Peas

&

```
LET &Ingredient2 = EXTRACT(&recipe,15,9)
```

if &recipe was

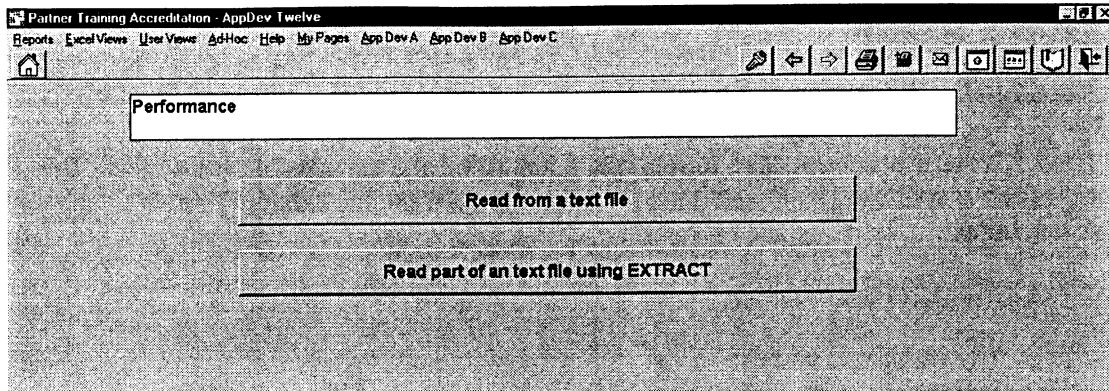
Chick Peas Olive Oil Tahini Lemon Juice Yoghurt Garlic

after using EXTRACT the variable &Ingredient2 would contain:

Olive Oil

Practical 3 - 2 Extract

Aim: To see part of a file using the EXTRACT command.



Guidelines

- Create a new routine.
- Read a data file into a variable
- Read a line from that file into another variable.
- Use "EXTRACT" to extract one word from the line variable.

Pseudo Code

ROUTINE Read IS

```
LET Variable1 BE READONLY FILE pathname;  
LET Variable2 = Variable1 nextLine;  
LET <Variable3> = EXTRACT(<Variable2>, "<start position for extract>", <no. of characters to be extracted>);  
LET EditText contents hold Variable3  
  
END;
```


SPLIT

The SPLIT function is used to split a text string into a number of sub-strings. The splitting is controlled by the position of separator characters on the original string, normally “,” or “|”.

SPLIT generates an array of the sub-strings and the number of sub-strings generated.

Four parameters are required:

- The variable to be split
- The split character
- The variable into which the resulting elements are to be stored.
- The variable into which the number of sub-strings is stored

```
LET <variable1> = SPLIT(<variable2>, "<split character>", <variable3>);
```

This splits *variable2* every time a *split character* is found.
The elements created by the split are loaded into the array *variable3* using an integer index.
The total number of entries is stored in *variable1*.

Example

```
LET &numberOfIngredients = SPLIT(&recipe, ",", &listOfIngredients);
```

if &recipe was

Chick Peas, Olive Oil, Tahini, Lemon Juice, Yoghurt, Garlic

after using SPLIT the variable array &listOfIngredients would contain:

Chick Peas
Olive Oil
Tahini
Lemon Juice
Yoghurt
Garlic

and &numberOfIngredients would = 6.

Read from a file using SPLIT:

```
#
#   Read an external file into an edit box using SPLIT
#
```

Pseudo Code

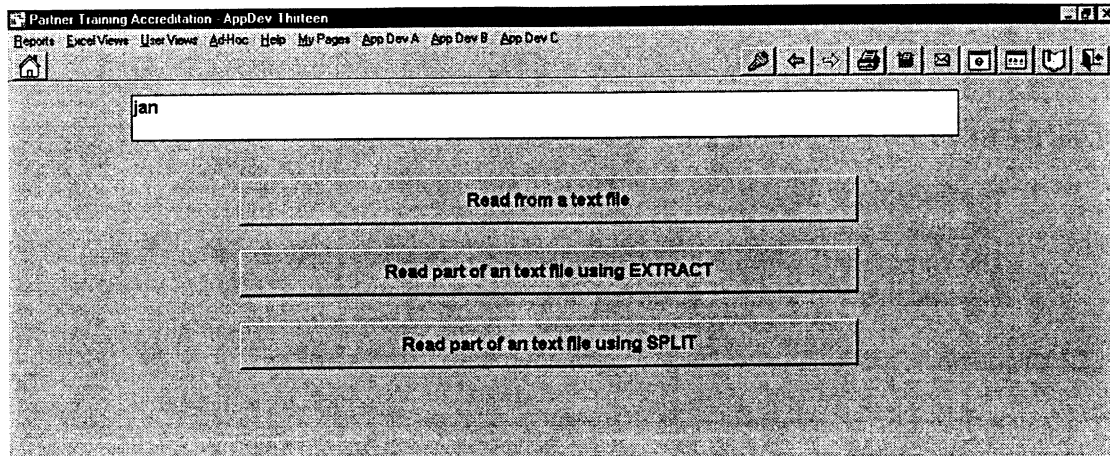
ROUTINE Read IS

```
LET Variable1 BE READONLY FILE pathname;
LET Variable2 = Variable1 nextLine;
LET <variable1> = SPLIT(<variable2>, "<split character>", <variable3>);
LET EditBox contents hold <variable3>
```

END;

Practical 3 - 3 Split

Aim: To see part of a file using the SPLIT command.

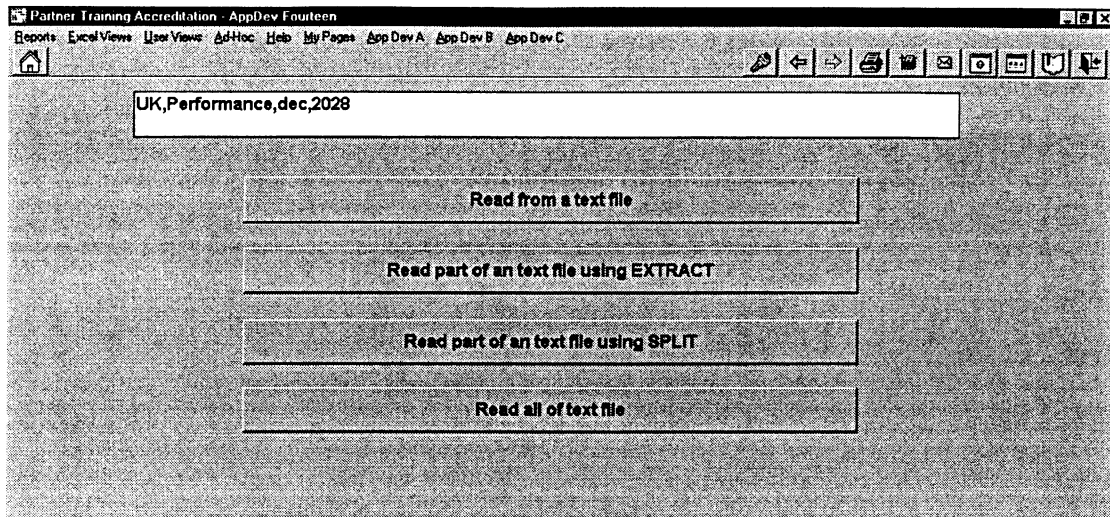


Guidelines

- Create a new routine.
- Read a data file into a variable
- Read a line from that file into another variable.
- Use "SPLIT" to extract one word from the line variable.
- Display the string from the first position in the file.
- Then display the number of instances within the array.

Practical 3 - 4 Loop Through a File

Aim: To read all of a file using a loop.



Guidelines

- Create a new routine.
- Read a data file into a variable
- Read a line from that file into another variable.
- Create a "FOR DO" loop.
- Drop out of the loop when the "endOfFile" is reached.
- Stop the loop using "BREAK" .
- While the loop is running the routine could be performing a number of operations on the current line of the file. These operations have to be held in a "BEGIN.....END" construction.

Pseudo Code

ROUTINE Read IS

LET Variable1 BE READONLY FILE pathname;

Start loop

LET Variable2 = Variable1 nextLine;

If end of file then BREAK the loop;

LET Edit Box contents hold Variable2

END;

Writing to a file

To write out from a variable into a text file, one first has to define and open the file, this is done using "NEW FILE" which will create (or overwrite if there is an existing file of that name) the required file. The reserved word FILE will also create a new file or append to one that already exists. Once the file has been created the command "PRINT ON" is used to push out text from Gentia to the file. Everything that is written to a file is written onto one line, so you have to tell the system to start a new line, the command for this is "NL". You have to close the file when you have completed writing to it. A modifier is used to close the file called "closeFile", it requires a Boolean value to set the file status, the value YES is used to close the file, and NO to indicate an open file.

for example

```
#      Write to an external file from an edit box
#

LET $file BE NEW FILE "C:\\GENTIA\\AD\\TEXT3.TXT";
IF $file error THEN
    CALL !_warningAlert(&message="Problem opening file")
FI;

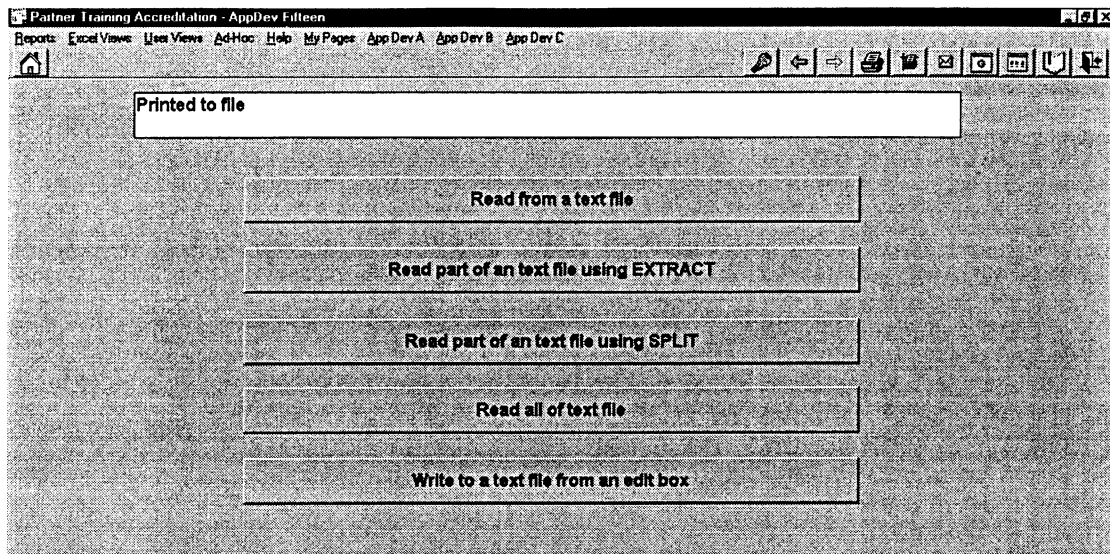
PRINT ON $file $wLine1Edt contents;
PRINT ON $file NL;

LET $file closeFile = YES

end;
```

Practical 3 - 5 Write to a File

Aim: To write to an external file.



Guidelines

- Create a new file
- Use "PRINT ON " to send a string to a file
- Use "NL" to force a carriage return
- Use "closeFile" to complete the process

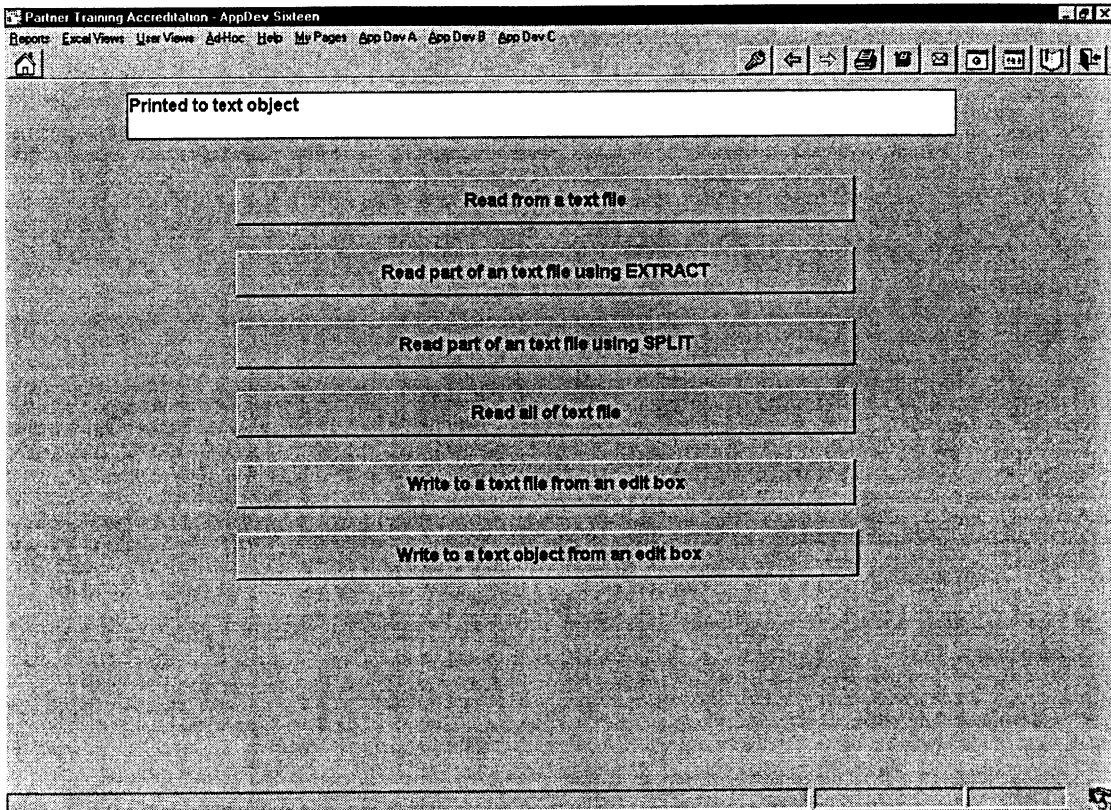
Pseudo-Code

routine seven is

```
file BE NEW FILE pathname;
PRINT ON file "abcde..";
PRINT ON file NL;
file closeFile = YES
```

Practical 3 - 6 Write to a Text Object

Aim: To write to a text object.



Guidelines

- Create a new text object or append to an existing one
- Use "PRINT ON " to send a string to a file
- Use "NL" to force a carriage return
- Use "closeFile" to complete the process

Pseudo-Code

routine seven is

```
file BE NEW TEXT pathname;
PRINT ON file "abcde..";
PRINT ON file NL;
file closeFile = YES
```


Gentia's Performance Impact Application

(ref: GPM01)

Copyright © Gentia Software plc 1999
All rights are reserved.

ACCURACY

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Gentia Software plc accepts no responsibility, and offers no warranty, whether expressed or implied, for the accuracy of this publication.

RESTRICTED RIGHTS

You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language or computer language, in any form or by any means, electronic, mechanical, optical, magnetic, photographic, manual or otherwise, any part of this publication without the express permission of Gentia Software plc.

This document refers to the products and tradenames of many manufacturers. Acknowledgement is made of all trademarks, registered trademarks, and trading names that are referred to in the text.

**Gentia's
PerformanceImpact
Application**

GENTIA™

© Copyright Gentia Software, 1999

Objectives

- ◆ To become familiar with the concepts of the application and database
- ◆ To be able to use PerformanceImpact to build an application
- ◆ To understand the implications of product customization

GENTIA

© Copyright Gentia Software, 1999

Agenda

- ◆ Introduction
 - EPM
 - Gentia Software
 - Multidimensionality
- ◆ Analysis with PerformanceImpact
- ◆ The PerformanceImpact Database
- ◆ Integration with Gentia
- ◆ PerformanceImpact on the web
- ◆ Administration and Customization

GENTIA™

© Copyright Gentia Software, 1999

Introduction

- ◆ The need for managing the performance of your business
- ◆ What is Gentia's PerformanceImpact application?
- ◆ What are the benefits?
- ◆ Multidimensionality

GENTIA™

© Copyright Gentia Software, 1999

Managing Business Performance

- ◆ Many factors make it difficult for businesses to succeed:
 - turbulent, competitive markets
 - mergers, acquisitions
 - regulatory changes
 - new delivery channels
 - more sophisticated customer expectations
- ◆ Organizations need to execute strategy to succeed
- ◆ Organizations need to be effectively positioned to execute strategy

GENTIA™

© Copyright Gentia Software, 1999

Tracking, measuring and managing business performance

Strategy is critical to guide organizations through turbulent markets and ever-changing competitive environments. It is also paramount to succeed in today's business world.

Mergers, acquisitions, outside entrepreneurial competition, regulatory changes, product and service convergence, new delivery channels, and more sophisticated customer expectations are some of the factors dictating organizations to change at an unprecedented pace. And, only those organizations effectively positioned to execute strategy and maximize business performance will succeed.

EPM - Enterprise Performance Management

- ◆ EPM solutions help drive business performance:
 - to promote corporate objectives and strategy
 - at strategic, management and operational levels
 - by analyzing, measuring and communicating
 - to generate conclusive, fact-based and actionable information
 - to produce strategic differentiators
 - that separate high growth businesses from those to be left behind.

GENTIA™

© Copyright Gentia Software, 1999

Enterprise Performance Management (EPM) solutions help drive business performance at strategic, management and operational levels by analyzing, measuring, and communicating information to promote corporate objectives and strategy.

Increasingly, organizations are turning to EPM solutions – systems that generate conclusive, fact-based and actionable information – as strategic differentiators that separate high-growth businesses from those soon to be left behind.

Gentia Software

- ◆ We have >10 years experience in helping many organisations to address their performance management requirements, eg:
 - Volvo
 - JP Morgan
 - McDonald's
 - Bell Canada
 - Fortis
 - Swiss Reinsurance

GENTIA™

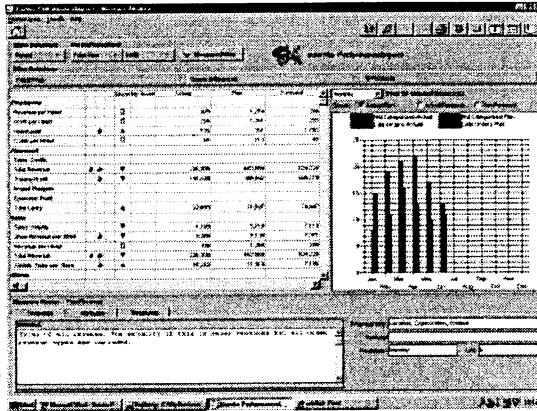
© Copyright Gentia Software, 1999

Gentia Software can help. With over a decade of experience in assisting organizations to implement EPM solutions, we know the business better than anyone.

Clients like JP Morgan and Company, Volvo, McDonald's Restaurants, Bell Canada, Fortis and Swiss Reinsurance have turned to Gentia to help address their most demanding performance management requirements.

PerformanceImpact

- ◆ Gentia's complete EPM solution
- ◆ Easy to set up and maintain
- ◆ Quick to customize
- ◆ Used to track, monitor and manage KPIs
- ◆ Fully client/server and web deployable to 1000's of users



GENTIA™

© Copyright Gentia Software, 1999

Gentia Software's packaged EPM solutions are easy to setup and maintain and can be quickly customized to meet precise requirements.

For tracking, monitoring and managing key performance indicators, Gentia PerformanceImpact is offered as a complete solution for measuring business performance enterprise-wide.

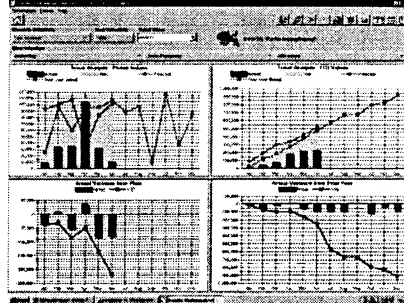
The solution is fully client/server and Web deployable to thousands of users enterprise-wide.

With its ability to analyze data buried in huge installed bases of ERP applications, Data Warehouses and legacy systems, a Gentia PerformanceImpact solution provides executives, managers and knowledge workers with information and insight necessary to capitalize on opportunities, resolve challenges and more effectively measure and manage business operations.

Gentia Software recognizes that measurement and management needs may change as business grows. An integrated platform is offered to take you from performance measurement all the way up to a complete strategy management system, such as the Balanced Scorecard. Gentia's integrated architecture makes this an easy and straightforward process by ensuring consistent user interfaces and seamless application integration.

Measurement & Analysis Functionality

- Track key performance indicators
- Set system-wide and user-defined performance thresholds
- Apply weightings to measures
- Conduct benchmark, performance and trend analysis with support for multiple measures
- Chart key indicators including Actual-to-Plan-to-Forecast
- Perform ad hoc analysis
- Send reminders to other individuals
- Attach comments to measurements
- Analyze data through filtering, slicing and dicing, and drill-down capabilities, including support for time dimensionality
- Define calculation rules, including automatic YTD calculations



GENTIA™

© Copyright Gentia Software, 1999

Gentia's PerformanceImpact solution delivers the functionality required for performance measurement and analyses, including the ability to:

- Track key performance indicators
- Set system-wide and user-defined performance thresholds
- Apply weightings to measures
- Conduct benchmark, performance and trend analysis with support for multiple measures
- Chart key indicators including Actual-to-Plan-to-Forecast
- Perform ad hoc analysis
- Send reminders to other individuals
- Attach comments to measurements
- Analyze data through filtering, slicing and dicing, and drill-down capabilities, including support for time dimensionality
- Define calculation rules, including automatic YTD calculations

Administrative Tools

- ◆ Gentia PerformanceImpact offers administrators a complete set of tools to:
 - set up & maintain the application
 - define measures
 - establish system thresholds
 - set weightings
 - determine metrics
 - define formulae
 - perform YTD calculations
 - and much more...

GENTIA™

© Copyright Gentia Software, 1999

Gentia PerformanceImpact offers administrators a complete set of tools to easily setup and maintain the application.

Administrators have full capabilities to define measures, establish system thresholds, set weightings, determine metrics, group objectives into perspectives, define formulae, perform YTD calculations, and much more.

Multidimensionality

- ◆ Defines the way that the information is held
- ◆ Allows the information to be accessed by the users:
 - intuitively
 - quickly
 - flexibly
 - easily
- ◆ But what does multidimensionality mean?

GENTIA™

© Copyright Gentia Software, 1999

Multidimensionality

Fruit

	B	A	O	P	G		
Unit Price	5.00	8.00	3.00	2.00	6.00		
Units Sold	200	400	300	800	600		

Measures

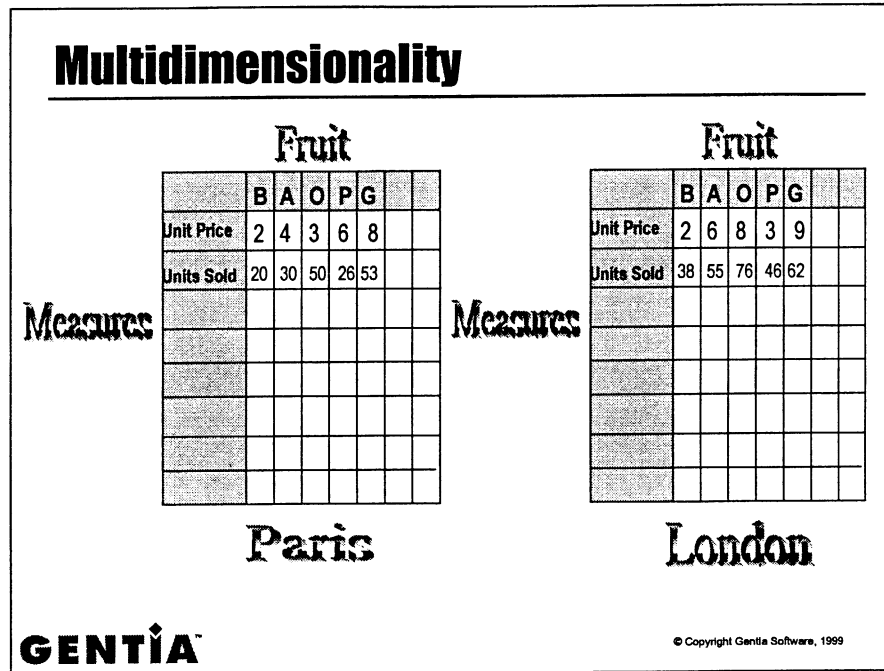
GENTIA

© Copyright Gentia Software, 1999

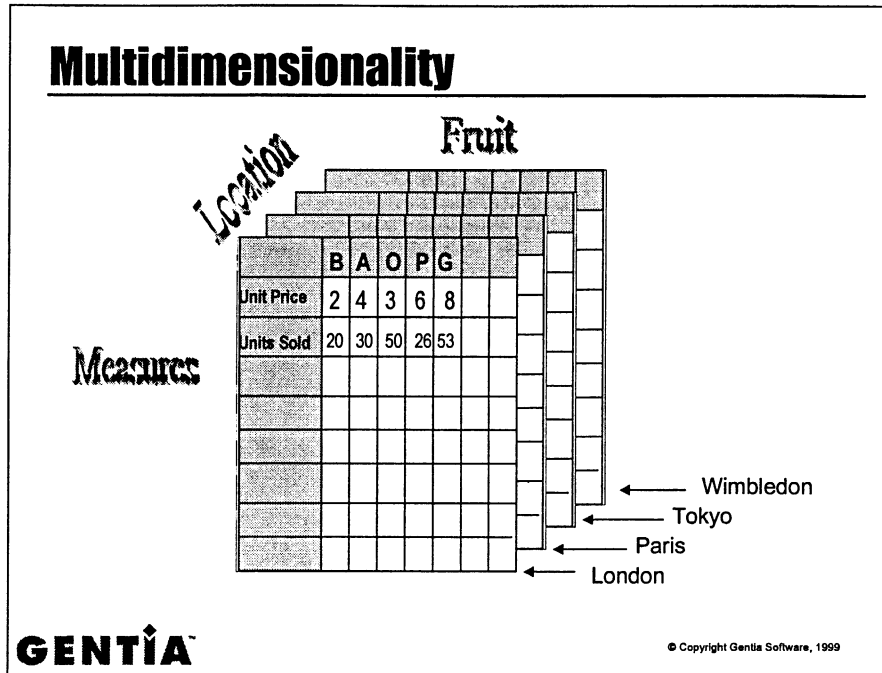
Data is stored by Gentia in a multidimensional manner. This means that it is possible to view the data from a number of different perspectives. First we will understand how the data is stored (and how multidimensionality works) and then we will see how the data is viewed from different angles.

In the two dimensional model above, which focuses on the sale of fruit for a green-grocery business, we can see some measures for **Banana, Pears, Grapes, Oranges** and **Apples** sold. The two measures are **Unit Price** and **Units Sold**. The measures are stored in the rows and the fruit is stored in the columns.

It is simple to represent data when only two dimensions are being used. What the above table doesn't show us however, is when the fruit was sold, where it was sold or who sold it. This would involve introducing a number of other dimensions.



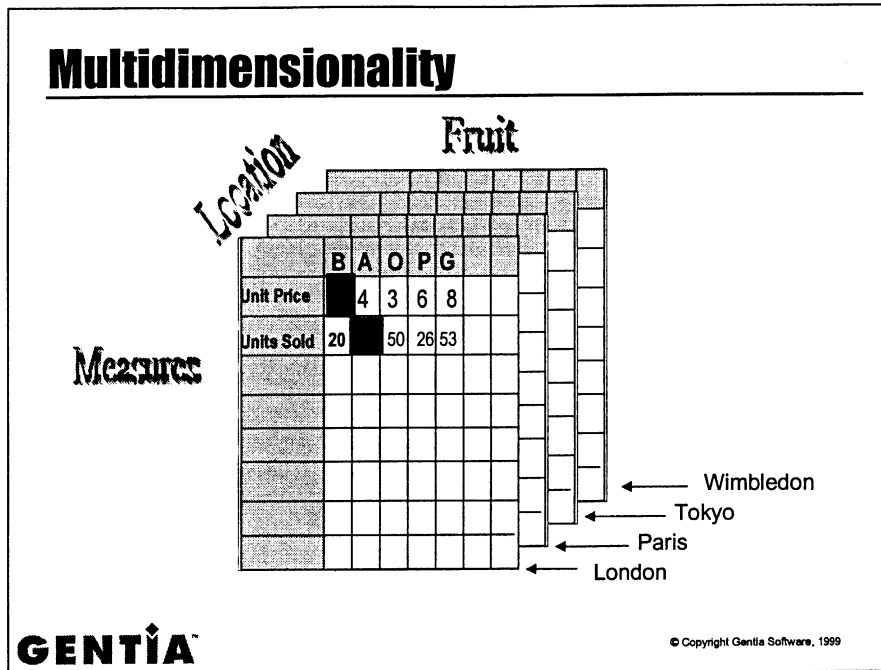
Multidimensionality comes into its element where three or more dimensions are being used. In the above example we have introduced the idea of a location element or dimension. We need to know the figures for Fruit sold in both London and Paris. The problem now arises of how to pictorially display the data. Two locations is fine but what if there were 100 locations. Obviously looking at 100 layouts would be too difficult.



It is perhaps easier to understand the third dimension if it is laid behind the first two. So we have **Location** as the third dimension giving us a number of different slices, one for each of the locations.

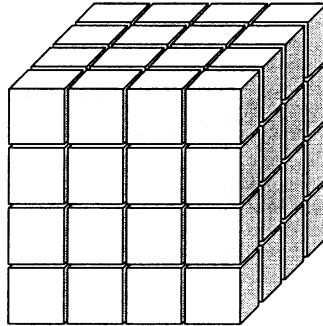
Each of the dimensions consists of a number of members. These are the components of the dimensions. The members for **Measures** are *Unit Price* and *Units Sold*. The members for **Fruit** are *Bananas*, *Apples*, *Oranges*...etc. These dimensions are laid out in the rows and columns and therefore all the data for their members is displayed. The third dimension, **Location** also has a number of members, *London*, *Paris*...etc but only one of these members' data is ever displayed at any one time.

What this means is that only the dimensions in the rows and columns ever show all their data. Any additional dimensions can only show one member at one time. The benefits of a multidimensional product is that the data can be viewed from any angle, meaning that any dimension can be moved into the row and column position. This is explained later.



With multidimensionality each figure of numerical data is associated with a number of dimensions. In the above example, there are two figures. One is for *Unit Price, Bananas and London*. The other is for *Units Sold, Apples and London*. This view of multidimensional data is displayed by Gentia in a cube form. The cube idea shows that all the data for fruit sold in all locations is contained in the cube but we can only view the data for the rows and columns. This is called a *slice* of data.

Data Viewed as a Cube

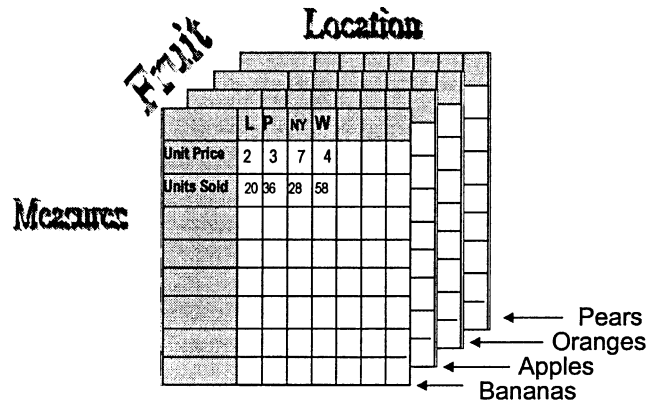


GENTIA™

© Copyright Gentia Software, 1999

The idea of using a cube is that there are many sides by which to view the data. Each cell which makes up the cube structure represents a dimension / member combination.

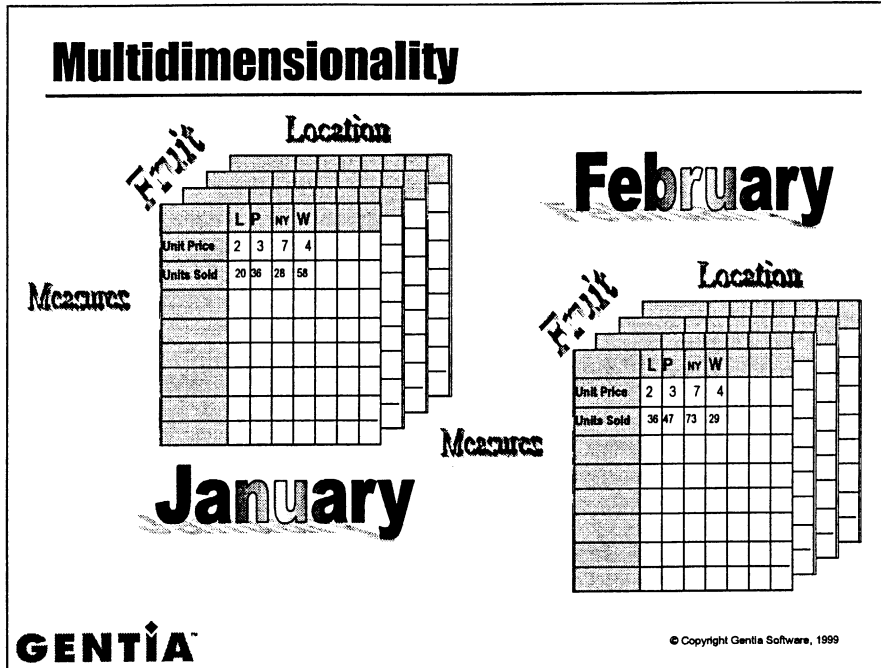
Same model, different view!



GENTIA

© Copyright Gentia Software, 1999

In the above example, the same model is still being accessed however, the model has been *turned around* so that the data now has Location as its horizontal axis and Fruit has become the depth axis. The advantage of this is that we can now compare the different measures for all our locations, referencing each fruit separately.



Of course, when there are many dimensions (**Time** has been introduced above) it becomes difficult and complicated to imagine it in pictorial format. The principle, however remains the same. When we look at a 'chunk' of data which represents a number of dimensions, we are looking at a 'slice' of data. The idea is that we are slicing into the cube to remove and view the relevant figures.

Mandatory Dimensions

- ◆ PerformanceImpact has the following dimensions with some default members already defined:
 - Measure - all defined by the user
 - Version
 - actual, plan, forecast, actual v. plan, plan variance %, act v. forecast, forecast variance %
 - Data View
 - period, year to date
 - Period
 - week01-week52, month01-month12, quarter01-quarter04
 - Year
 - 1998 - 2002

GENTIA™

© Copyright Gentia Software, 1999

The measures can be calculated from other measures.

The measures can belong to user-defined categories, used to group related measures together; a measure can belong to more than one category.

More on this later.

Multidimensionality

- ◆ Allows many data types or dimensions
- ◆ A number of dimensions create a 'slice'
- ◆ Dimensions contain many members
- ◆ Allows drill up / down

GENTIA

© Copyright Gentia Software, 1999

In short, multidimensionality addresses the problem of only being able to view data in row and column (and sometimes depth) format. Rows and Columns work well with two or even three dimensions, but fall down when many dimensions are used. To reduce say, 7 dimensions into a two dimensional pictorial format is impossible. We must know which dimensions we are looking at, besides the two that are pictorially represented. This information can then be changed and re-arranged to suit the user's needs and wishes.

It is important to remember that the dimensions displayed in the rows and the columns will display all their data, but the slice information provided by the other dimensions shows only data from one member within that dimension. This slice information is displayed either in a Gentia Listbar, or on buttons attached to the page view.

Chapter 1

Using the PerformanceImpact
Application for Analysis

GENTIA™

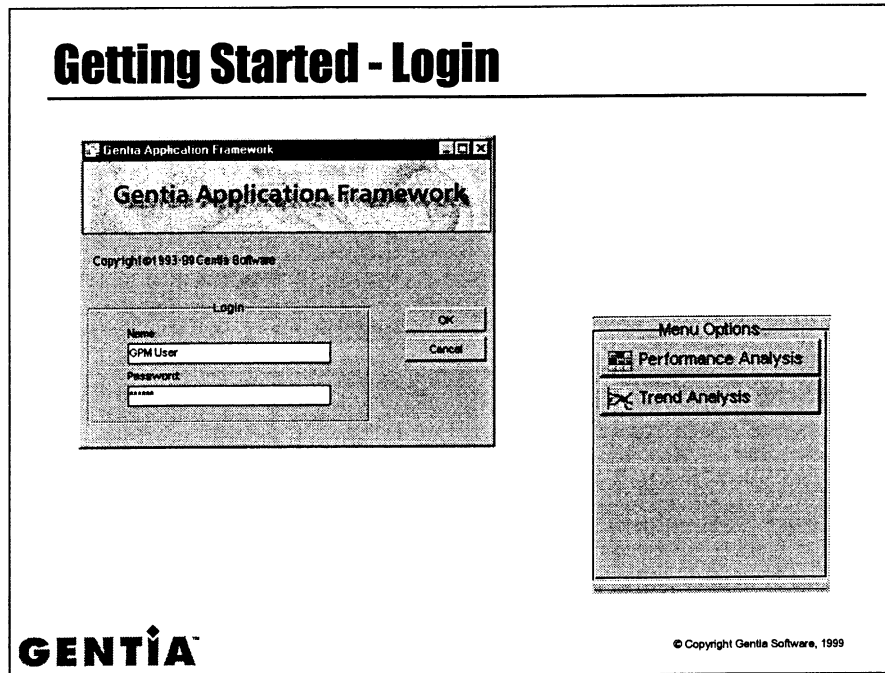
© Copyright Gentia Software, 1999

What We Will Cover...

- ◆ Getting started - login
- ◆ Menu system
- ◆ Measure analysis
- ◆ Measure filter
- ◆ Version trend analysis
- ◆ Measure trend analysis
- ◆ Ad-hoc analysis
- ◆ Page customization

GENTIA™

© Copyright Gentia Software, 1999



The PerformanceImpact application runs client/server, so assuming that the server is running, you can start the application from the Windows *Start* menu.

You will be presented with the *Login* window where you'll be prompted for your username and password.

Clicking on *OK* will take you to the default page, which is usually defined to be the *Measure Analysis* page.

The Menu is displayed in the top left hand portion of the home page and is used to move to the other analysis pages. There are 4 analysis pages:

- Measure Analysis
- Version Trend Analysis
- Measure Trend Analysis
- Ad-hoc Analysis

The menu options match the drop down options which are permanently displayed across the top of every screen.

Measure Analysis


View Selections: Period Selections: Total Year: 1998 Measure Filter


Site Selections: World Wide Cosmo Enterprises All Products


	C	S	W	Score My Score	Actual	Plan	Forecast	Act v Plan	Plan Var
Employee									
Revenue per Head					307.9	1,264.0	297.5	-566.04	
Profit per Head					254.2	1,050.8	255.3	-658.07	
Headcount					735.0	351.0	1,762.0	334.00	
Costs per Head					53.7	213.2	42.3	92.00	
Financial									
Sales Credits					0.0	0.0	0.0		
Total Revenue					226,308.0	443,669.0	524,225.0	-217,361.00	
Trading Profit					186,820.0	368,842.0	449,778.0	-252,700.00	
Project Budgets					0.0	0.0	0.0		
Economic Profit					0.0	0.0	0.0		
Total Costs					39,488.0	74,827.0	74,447.0	35,339.00	
Sales									
Sales Volume					4,793.0	5,718.0	7,110.0	-925.00	
Shoe Revenue per Store					5,389.0	8,134.0	7,371.0	-2,745.00	
Revenue per Head					307.9	1,264.0	297.5	-566.04	
Total Revenue					226,308.0	443,669.0	524,225.0	-217,361.00	
Athletic Sales per Store					16,242.0	11,918.0	7,185.0	4,324.00	
Stores									
Number of Stores					105.0	62.0	97.0	43.00	
Shoe Revenue per Store					5,389.0	8,134.0	7,371.0	-2,745.00	
Athletic Sales per Store					16,242.0	11,918.0	7,185.0	4,324.00	


GENTIA © Copyright Gentia Software, 1999

The *Measure Analysis* page will display the measures in the rows, grouped in categories. The columns are a mixture of indicators, values and variances.

The *Comment Indicator*  in the C column indicates a comment is present. A user can add or view a comment for a measure or period. If no paperclip is present, clicking on a blank cell will allow the user to add a fresh comment.

The *RBSC Indicator*  in the S column indicates that the measure is also defined in the RBSC application. Clicking on this icon will allow the user to drill through into the RBSC application.

The *Weighted indicator*  in the W column indicates that this measure has been created using weighted versions of other measures.

The *Score*  reveals the score for the measure, relative to the system thresholds.

As well as the system thresholds for each measure, thresholds can be defined by the user and are displayed in the *My Score* column. Users have the ability to view, edit or delete the thresholds for any measure if they so wish.

Measure Analysis

View Selections: Period Selections: Total Year: 1999 Measure Filter

Slice Selections: World Wide Cosmo Enterprises All Products

	C	S	W	Score/My Score	Actual	Plan	Forecast	Act v Plan	Plan Var
Employee									
Revenue per Head				<input type="checkbox"/>	307.9	1,264.0	297.5	-956.04	
Profit per Head				<input type="checkbox"/>	254.2	1,050.8	255.3	-796.67	
Headcount				<input checked="" type="checkbox"/>	735.0	351.0	1,762.0	334.39	
Costs per Head				<input type="checkbox"/>	53.7	213.2	42.3	92.03	
Financial									
Sales Credits					0.0	0.0	0.0		
Total Revenue	0	↓		▼	226,308.0	443,669.0	524,225.0	-217,361.00	
Trading Profit		↓		▼	166,820.0	368,842.0	449,778.0	-282,958.00	
Project Budgets					0.0	0.0	0.0		
Economic Profit					0.0	0.0	0.0		
Total Costs					39,488.0	74,827.0	74,447.0	35,339.00	
Sales									
Sales Volume				▼	4,793.0	5,718.0	7,110.0	-925.00	
Shoe Revenue per Store		↓		▼	5,389.0	8,134.0	7,371.0	-2,745.00	
Revenue per Head				▼	307.9	1,264.0	297.5	-956.04	
Total Revenue	0	↓		▼	226,308.0	443,669.0	524,225.0	-217,361.00	
Athletic Sales per Store		↓		▲	16,242.0	11,918.0	7,185.0	4,324.00	
Stores									
Number of Stores				▲	105.0	62.0	97.0	43.00	
Shoe Revenue per Store		↓		▼	5,389.0	8,134.0	7,371.0	-2,745.00	
Athletic Sales per Store		↓		▲	16,242.0	11,918.0	7,185.0	4,324.00	

GENTIA™ © Copyright Gentia Software, 1999

The *Measure Analysis* page has the following functionality:

- Change the slice of data, by using the *Slice Selections* listbar above the table.
- Use the *View Selections* button to switch between the ytd and period data view.
- Use the *Period Selections* button to change the year and period for which you wish to view the data.

Further functionality is discussed in the next few pages.

Measures

- ◆ Each measure - belongs to a category
 - e.g. sales, financial,
- ◆ Each measure - associated with dimensions
 - e.g. *unit price* only applies to product and time, and possibly version
 - e.g. *overheads* does not apply to product
 - Not for calculated or weighted measures
- ◆ Each measure - has a periodicity
 - i.e. monthly, quarterly, yearly

GENTIA™

© Copyright Gentia Software, 1999

Each measure can belong to one or more categories to provide a means of grouping related measures. A measure can belong to more than one category.

Dimensions, as we have already seen provides a way of identifying your data

e.g. location, division, product, time

The measures can be associated with dimensions. For example:

The measure *unit price* of something you sell may only apply to the product, time and perhaps version dimensions, since it will be universal throughout the location and division dimensions

The measure *overheads* may not apply to the product dimension, since they are concerned with the office locations and does not apply at individual product level

A measure will also have a periodicity associated with it, either monthly, quarterly or yearly.

Measure Filter

GENTIA™

© Copyright Gentia Software, 1999

By default, all measures will be displayed on the *Measure Analysis* page, but you can define exactly which measure you wish to view.

In the *Measure Analysis* page, clicking on the *Measure Filter* button will open this window where you can specify which measures are to be displayed in the rows.

Uncheck the *Show All Measures* box and apply the functionality as follows:

Select by Category - to view measures by category

Select by Dimension - to view measures according to the dimensions to which they apply

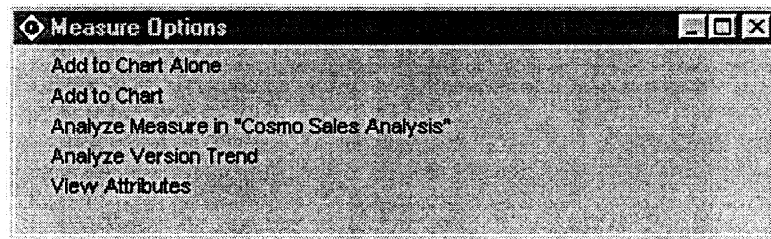
The *Show measures where ANY dimension applies* checkbox can be used for more general dimension selections

Select by Periodicity - to view measures according to their periodicity

Select by Type - to view measure according to whether they exist in the RBSC application and whether they are weighted.

Measure Analysis

- ◆ By clicking on the name of the measure or on one of its values:

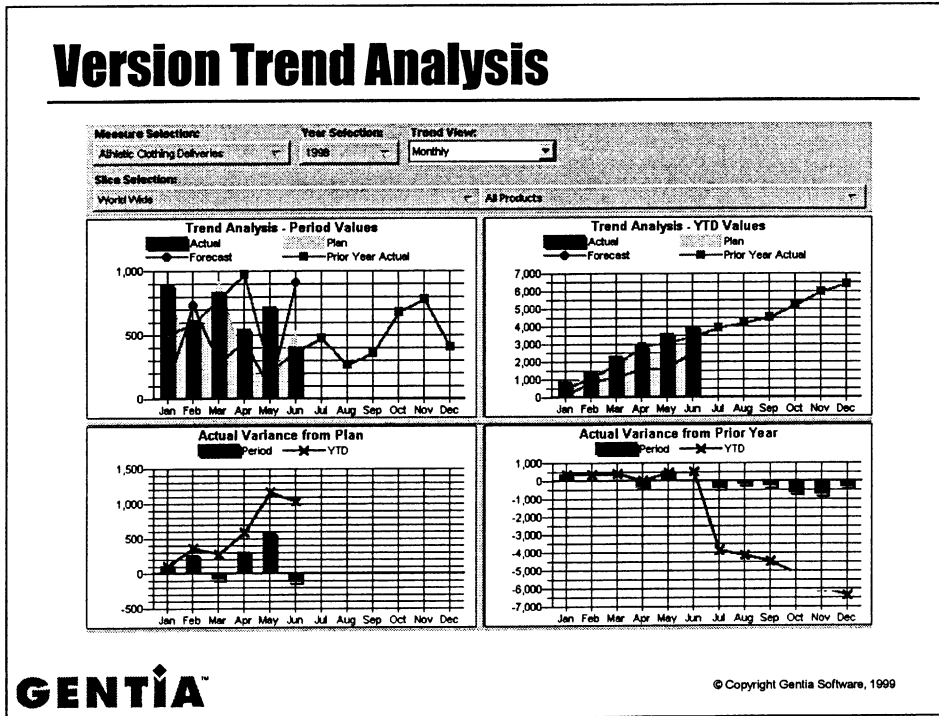


GENTIA™

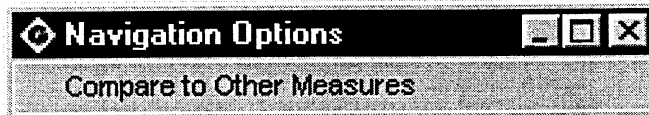
© Copyright Gentia Software, 1999

Simply click on the measure name or on one of its values, for the following options:

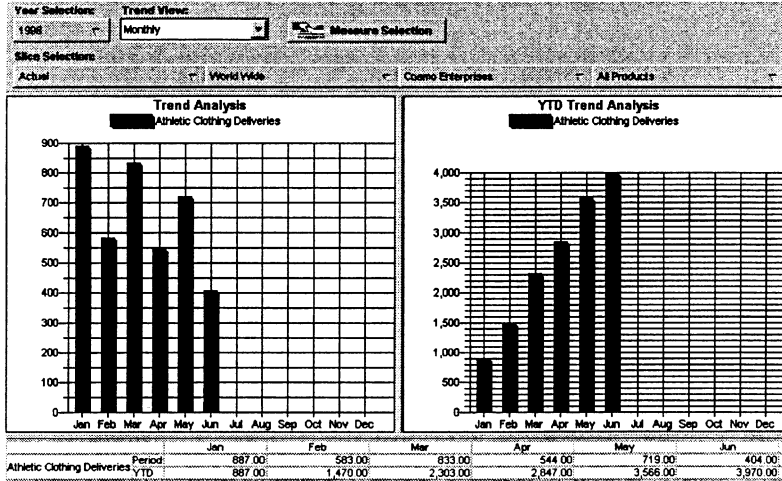
- *Add to Chart Alone* - displays a trend chart for the selected measure.
- *Add to Chart* - adds another measure to the trend chart.
- *Analyze Measure in [application name]*. This option only appears if there is an associated drill through application.
- *Analyze Version Trend* - described later in this chapter.
- *View Attributes* - information about this measure - see later.



- Compares multiple versions of a single measure over time
- Shows **actual** and **variance** charts
- Can *Compare to Other Measures*



Measure Trend Analysis

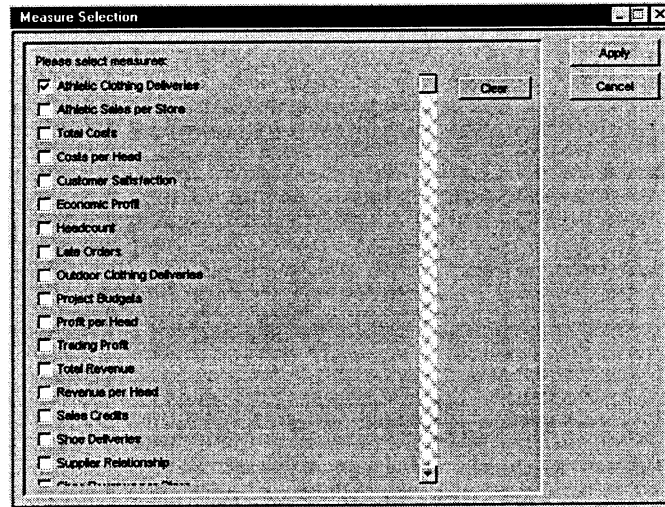


GENTIA™

© Copyright Gentia Software, 1999

- Compares the performance of a number of measures
- Compared over time
- Period data and YTD performance

Measure Selection



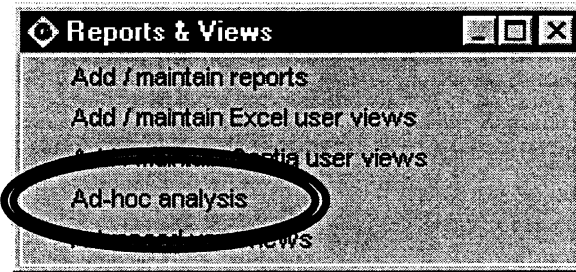
GENTIA™

© Copyright Gentia Software, 1999

Measure Selection allows the user to select more measures or different measures for comparison. Simply select from the list and click *Apply*.

Ad-hoc Analysis

- ◆ from the framework menu



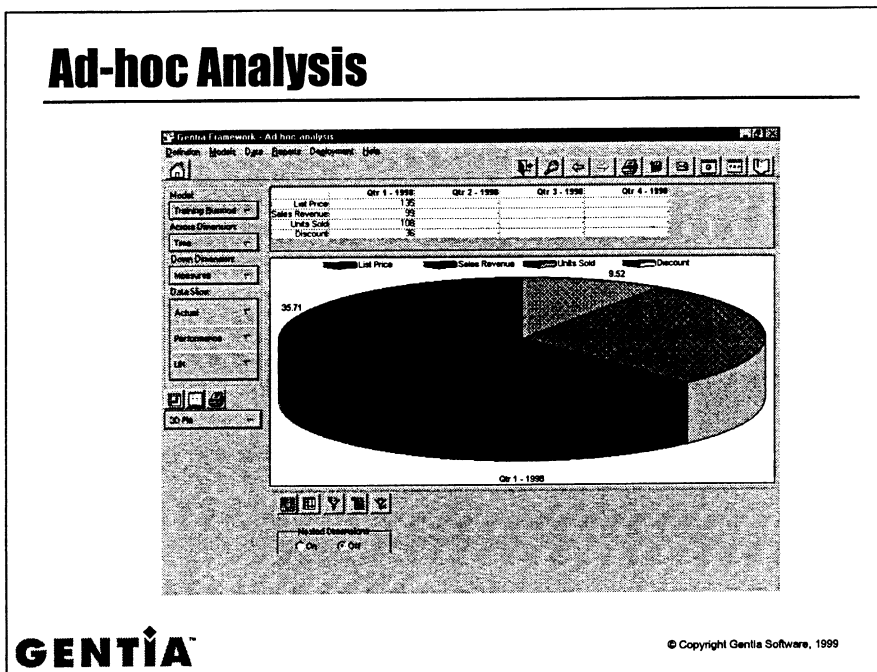
GENTIA™

© Copyright Gentia Software, 1999

Ad-hoc Analysis provides a facility for you to view data in an application.

Simply select the option and a user will be presented with a number of ways of looking at the data, table or chart. The user can also print, add textual comments and re-order data by sorting, suppressing zeros and switching hierarchies on and off.

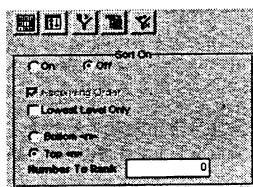
Ad-hoc Analysis



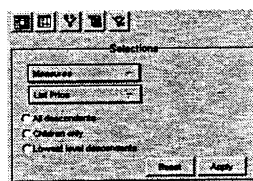
GDL has been used to assess which model specs or EIS models and scenarios the user has access to and it will create a list. Once this list has been compiled, the user can select any entry to be used as the data source (model).

The page is divided into three (using a split view technique) which displays a chosen model as a table and a chart. Options are displayed at the bottom of the page, allowing the user to modify their view. *Nested Dimensions* (above) will alter the display accordingly. In addition, there are the following:

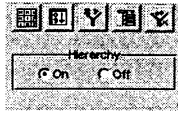
Sorting, which allows the user to sort records according to certain criteria.



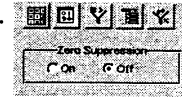
Selection of row/column dimension.



Hierarchy selection

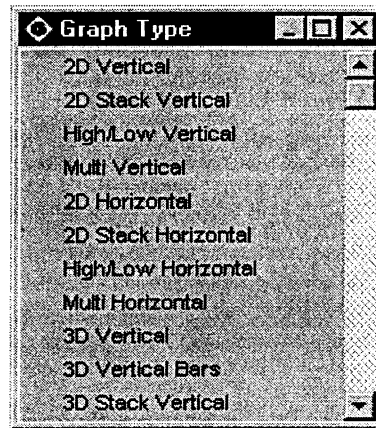
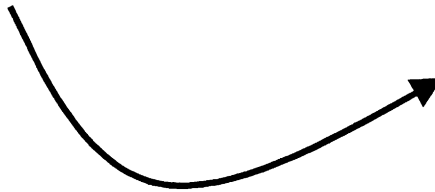


and zero suppression.



The left hand buttons allow the user to pivot the rows and columns, create text commentary for a printed report and print the report.

In addition, it is possible to change the appearance of the chart by simply selecting a new type.

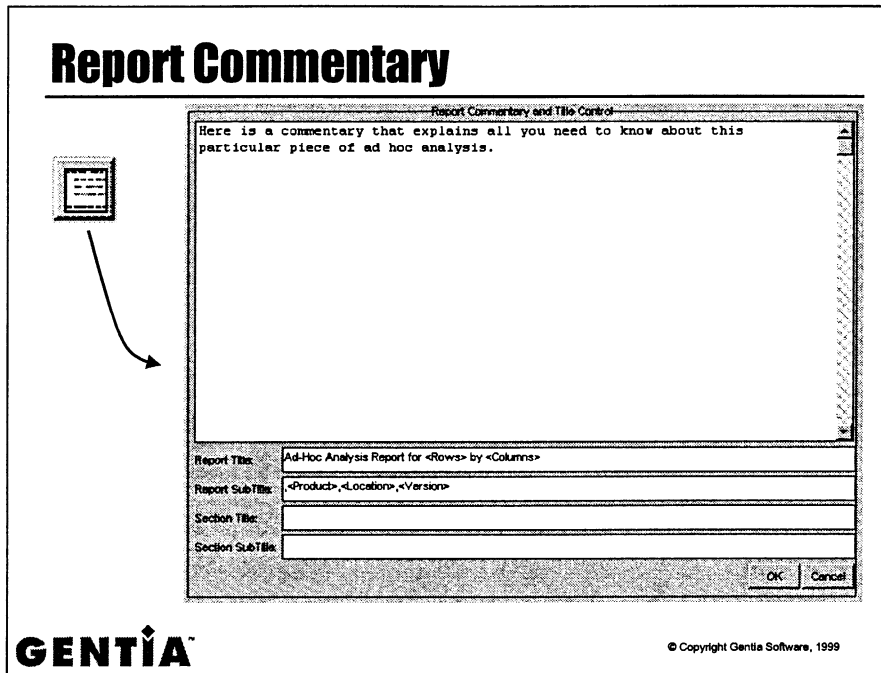


Menu System

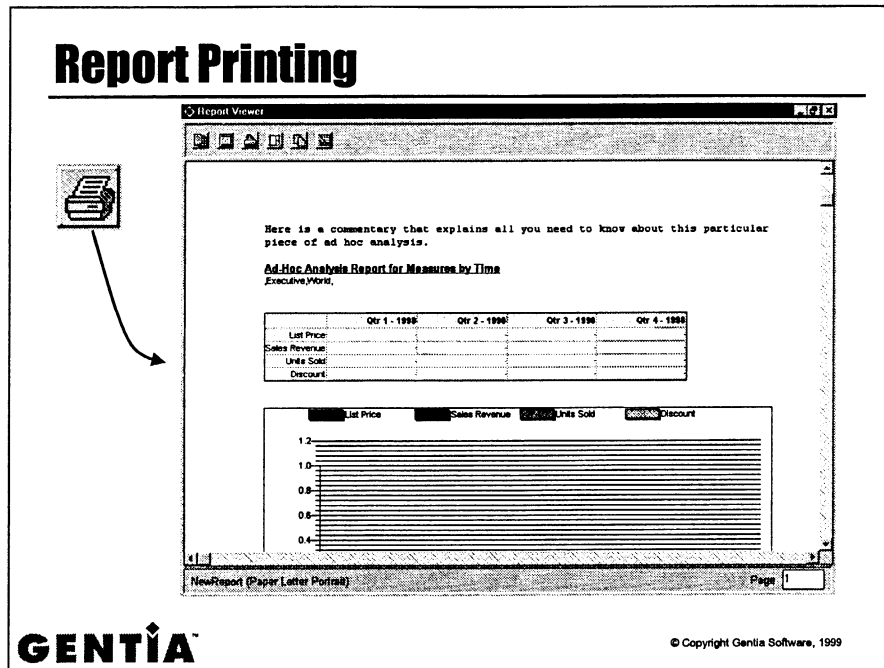
- ◆ Navigate to other applications and pages in four different ways
 - from PerformanceImpact menu buttons (top left)
 - use Gentia Applications button (on Gentia menu)
 - use Gentia bookmarks (on Gentia menu)
 - pop-up menu options (on *some* PerformanceImpact pages)

GENTIA™

© Copyright Gentia Software, 1999



Add associated commentary about the report.



Selecting the *Print* button takes the user into the *Report Viewer*. This is Gentia's own report and printing package and is covered in detail in the *Report Builder* course. At its simplest, it lays the table and chart on a page with any commentary inserted at the top of the page. A header or footer can be added and the report can be saved to disk.

The user can then select the layout and size of the paper and print the report.

Page Customization

- ◆ Standard pages can be modified
- ◆ Disable pages not required
- ◆ Extra pages can be added
- ◆ GDL knowledge is required

GENTIA™

© Copyright Gentia Software, 1999

Any of the standard pages can be modified or disabled to meet customer requirements. The Gentia Framework administrator is required to do this. Pages can be added to the application but a full knowledge of Gentia GDL and page building is required. These topics are covered on the *Building Gentia Pages* course and the *Gentia Application Development* course. In addition, the developer would need an understanding of the relevant PerformanceImpact GDL modules. This information is provided in the *Gentia PerformanceImpact Technical Document*.

Summary of Chapter

- ◆ Navigation
 - moving around the relevant PerformanceImpact pages.
- ◆ Pages analysis
 - analysing the various pages available
- ◆ Ad-hoc analysis
 - using the Framework's ad-hoc analysis
- ◆ Page Customization
 - How the pages are changes to meet customer requirements.

GENTIA™

© Copyright Gentia Software, 1999

Chapter 2

The PerformanceImpact Database

GENTIA™

© Copyright Gentia Software, 1999

What we will cover...

- ◆ Mandatory dimensions
- ◆ Optional Dimensions
- ◆ Relational tables

GENTIA™

© Copyright Gentia Software, 1999

The Performance Impact Database

- ◆ Measures and Calculated Measures
- ◆ Categories
- ◆ Dimensions
- ◆ Base and Join models

GENTIA™

© Copyright Gentia Software, 1999

The application will contain a number of measures and some of these will be calculated. The measures can be grouped into categories. These categories are defined by the user. Each measure will have user-defined dimensionality. The user will define a number of dimensions but not all dimensions must apply to all measures.

Each group of measures with the same dimensionality are grouped into the same base model. The base models are then combined using a join model.

Mandatory Dimensions

- ◆ These dimensions MUST appear in all PerformanceImpact applications
 - Measure
 - Version
 - DataView
 - Period
 - Year
- ◆ Can rename, remove or add members to these dimensions

GENTIA

© Copyright Gentia Software, 1999

Although all these dimensions are necessary, the members they contain can be modified according to each application. Each dimension is detailed below.

Measure - contains no pre-defined members, all members are user-defined. This dimension can contain calculated measures and user-defined measure categories.

Version - contains default members: *Actual, Plan, Forecast, Act v Plan, Plan Var%, Act v Fcst* and *Fcst Var%*.

DataView - contains default members: *Period* and *YTD*.

Period - contains default members: *W01 - W52, M01 - M12, Q01 - Q04*.

Year - contains default members: *1998 - 2002*.

Optional Dimensions

- ◆ As many as the user requires
- ◆ Linear or Hierarchical
- ◆ Loaded from ASCII files or SQL sources

GENTIA™

© Copyright Gentia Software, 1999

Base and Join Models

- ◆ Base models automatically created
 - contains measures with common dimensionality
- ◆ Automatically named

- ◆ Join models automatically created
- ◆ Contains weighted and calculated measures
- ◆ Model spec will be created
 - called GPM View

GENTIA™

© Copyright Gentia Software, 1999

Base Models

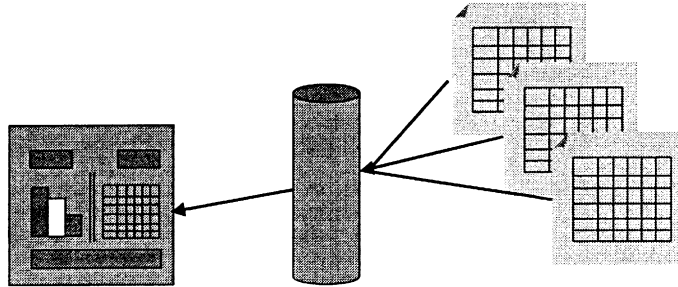
The model name is constructed automatically by concatenating TIME with the codes of the optional dimensions that have been included. For example, if there were two extra dimensions called Product and Location, the base model would be called Product and Location, the base model would be called TIME_PRD_LOC.

Join Models

A join model called **MEASUREView** is automatically created to join all base models.

Relational Tables

- ◆ Relational tables are used to contain specific data in addition to the GentiaDB



GENTIA™

© Copyright Gentia Software, 1999

The following information is stored in the tables:

- Measure definitions
- Category definitions
- Links between categories and measures
- Dimension definitions
- Dimension member definitions
- System and user-defined threshold values
- Pointers to the Gentia text objects that contain comments about measures
- GentiaDB model names
- The status of the GentiaDB Business Model

Measure Attributes

- ◆ Each measure has a number of attributes
- ◆ Setup by the PerformanceImpact administrator
- ◆ Use the Measure Administration page

GENTIA

© Copyright Gentia Software, 1999

Attribute	Notes
Name	Descriptive name for use on analysis pages
Code	Code for internal use
Unit	Narrative
Periodicity	Weekly/ Monthly/ Quarterly
Definition	Narrative
Intent	Narrative
Application Name	Name of the drill through application (optional)
Application Report	Name of the drill through application page (optional) <i>Use FWK: as a prefix if the page belongs to a Framework app</i>
Dimensionality	One or more of the user-defined optional dimensions <i>Calculated and weighted measures do not have any optional dimensions</i>
Categories	One or more of the user-defined categories (specified using the Category Administration page)
System Thresholds	Low and high % variance limits for color coded performance indicators <i>User thresholds are defined using the Measure Analysis page</i>

RBSC parameters	For measures that are also defined in RBSC, the names of the corresponding RBSC scorecard, objective and measure
Calculation type	Not calculated/ Calculated/ Weighted
Expression	Rule for calculation <i>The syntax must be correct although it is not validated on input</i>
Formula	Narrative description of calculation rule
Consolidation option	None (no consolidation at all) / Full (consolidation in all dimensions) / Hierarchical: Exclude Time (consolidation in all dimensions except Time) / Full: Last Period (consolidate in the Time dimension only, carrying forward the last loaded value) <i>None is used for all calculated and weighted measures</i>
Reverse Sign on Variance	This may be required for certain measures such as cost items

Summary

- ◆ Mandatory Dimensions
 - every PerformanceImpact database MUST include these
- ◆ Optional Dimensions
 - as many as the user requires
- ◆ Base and Join models
 - automatically created
- ◆ Relational Tables
 - contains additional information

GENTIA™

© Copyright Gentia Software, 1999

Chapter 3

Integration with Gentia

GENTIA™

© Copyright Gentia Software, 1999

Integration with Gentia products

- ◆ Drill through to/from RBSC
- ◆ Drill through to Gentia Analytical Applications
- ◆ PerformanceImpact Administration
- ◆ Getting started

GENTIA™

© Copyright Gentia Software, 1999

Drill through to/from RBSC

- ◆ RBSC measures can be analyzed in greater detail
- ◆ Drill down into the RBSC application
- ◆ Can drill into PerformanceImpact from other products
- ◆ Icon indicates which measures have corresponding RBSC measure

GENTIA™

© Copyright Gentia Software, 1999

The third column on the Performance Analysis page might contain an icon. The presence of an icon in the third column indicates that the measure can be drilled down on. This will launch the RBSC application (if present) and the measure can be viewed in far greater detail.

Drill through to Gentia Analytical Apps

- ◆ Can access ANY Gentia Analytical Application
- ◆ Can perform further analysis
- ◆ Can drill through to a FWK application

GENTIA™

© Copyright Gentia Software, 1999

Any Gentia Analytical Application can be accessed from the PerformanceImpact application. Each PerformanceImpact measure will need to have an associated drill through application name and an associated drill through page set up. If the drill through application is the framework, it will need to have a prefix of **FWK:** before the name, eg. FWK:ABC Analysis Page.

Drill through only works if the relevant application has been installed correctly. In addition, if it is a framework application it must be correctly defined within the current framework.

PerformanceImpact Administration

- ◆ Supports creation and maintenance of a PerformanceImpact database
- ◆ Can only be used by *Admin* and *GPMAAdmin*
- ◆ Functions are:
 - Dimension Administration
 - Measure Administration
 - Category Administration
 - Model Administration
 - BMScript Administration

GENTIA™

© Copyright Gentia Software, 1999

The details of the Administration role are explained within chapter 5.

Chapter 4

Performance Impact on the Web

GENTIA™

© Copyright Gentia Software, 1999

About This Section

- ◆ A demo of PerformanceImpact on the web
- ◆ Logging in
- ◆ Page Structure
 - Other applications
 - Performance analysis
 - Actual/Plan comparisons
 - Performance Trend
 - Ad-Hoc analysis
- ◆ Navigation and Drill Through

GENTIA™

© Copyright Gentia Software, 1999


One of the following Web browsers should be used:

- Internet Explorer v4.xx
- Sun's HotJava Browser v1.1.2
- Netscape Navigator v4.xx

Administration pages are not available on the web. These can only be accessed through the Gentia client.

Logging In

Gentia Application Framework



Name:
GPM User

Password:

GENTIA™ © Copyright Gentia Software, 1999

The web browser should be started and the correct URL should be entered into the *Address* line. This will be provided by the instructor or your web administrator and will be in the form of **<http://webserver/gpm.html>**.

The screen is divided into a number of frames. At a later stage these frames will contain extra relevant data. Enter the *Name:* and *Password:* and hit enter. The default page will be loaded. If no default page has been specified, the *Performance Analysis* page will be shown.

The default page is selected from the Application Framework. This is covered on the *Introduction to Gentia* course.

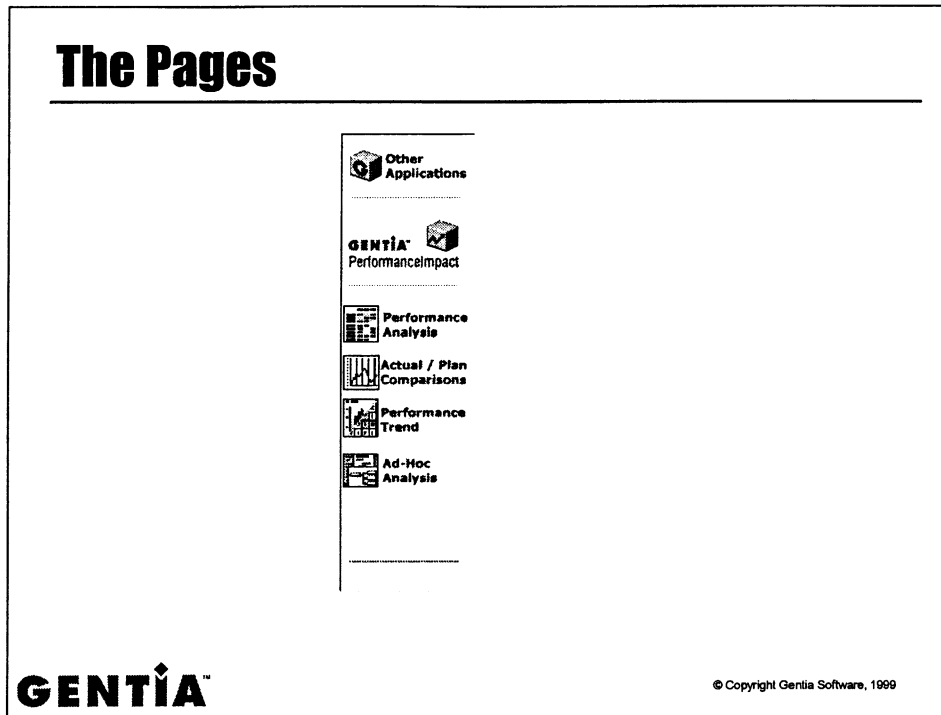
Page Structure

The screenshot displays the Gentia Performance Impact application interface. On the left is a navigation menu with options: Other Applications, Performance Analysis, Actual / Plan Comparisons, Performance Trend, and Ad-Hoc Analysis. The main area shows a data table for performance metrics, filtered by 'World Wide', 'Cosmo Enterprises', and 'All Products' for the year 1998. The table includes columns for C, S, W, Score, My Score, Actual, Plan, Forecast, Act v Plan, and Plan Var %.

	C	S	W	Score	My Score	Actual	Plan	Forecast	Act v Plan	Plan Var %
Employees										
Revenue per Head				□		307.9	1,264.0	297.5	-566.04	-0.45
Profit per Head				□		254.2	1,050.8	255.3	-658.07	-0.88
Headcount				↓	▲	735.0	351.0	1,762.0	384.00	109.40
Costs per Head				□		53.7	213.2	42.3	32.03	6.43
Financial										
Sales Credits						0.0	0.0	0.0		
Total Revenue	0	↓		▼		226,308.0	443,669.0	524,225.0	217,361.00	-48.99
Trading Profit				↓	▼	186,620.0	368,842.0	449,778.0	252,700.00	-96.22
Project Budgets						0.0	0.0	0.0		
Economic Profit						0.0	0.0	0.0		
Total Costs						39,488.0	74,627.0	74,447.0	35,339.00	47.23

© Copyright Gentia Software, 1999

Once the user has logged in, the menu bar appears in the left hand frame. This allows complete navigation to all the available pages within PerformanceImpact. The split view divider can be dragged to change the size of the page and the toolbar.



Other Applications allows the user to navigate from the PerformanceImpact to another application. These applications will appear on a pop up list. Only the applications which the user is allowed access to will appear. If the user has no access to other applications, the button will not be present.

Using the drill through method will also take a user to another application. This is explained in greater detail later.

Performance Analysis

Period	Total Year	1998	Measure Filter Index	Refresh Date
World Wide	Cosmo Enterprises		All Products	

	C	S	W	Score	My Score	Actual	Plan	Forecast	Act v Plan	Plan Var %
Employees										
Revenue per Head					□	307.9	1,264.0	297.5	-566.04	-0.45
Profit per Head					□	254.2	1,050.8	255.3	-658.07	-0.88
Headcount	↓				▲	735.0	351.0	1,752.0	384.00	109.40
Costs per Head					□	53.7	213.2	42.3	92.03	0.43
Financial										
Sales Credits						0.0	0.0	0.0		
Total Revenue	0	↓			▼	226,308.0	443,689.0	524,225.0	-217,361.00	-48.99
Trading Profit	↓				▼	186,820.0	368,842.0	449,778.0	-252,700.00	-96.22
Project Budgets						0.0	0.0	0.0		
Economic Profit						0.0	0.0	0.0		
Total Costs						39,488.0	74,827.0	74,447.0	35,339.00	47.23
Sales										
Sales Volume					▼	4,793.0	5,718.0	7,110.0	-925.00	-16.18

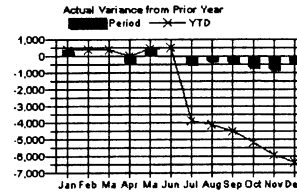
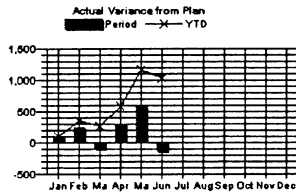
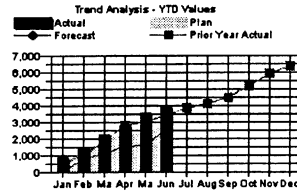
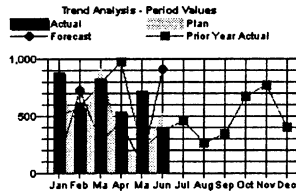
GENTIA

© Copyright Gentia Software, 1999

The *Performance Analysis* page is the same as the *Measure Analysis* page which is described earlier. The columns allow the user to drill up/down on the measures. In addition, the user can drill through to other Gentia applications by clicking on an icon in one of the columns.

Performance Trend

Athletic Clothing Deliveries | 1998 | Monthly | Refresh Data
World Wide | All Products
Compare with other measures



GENTIA™

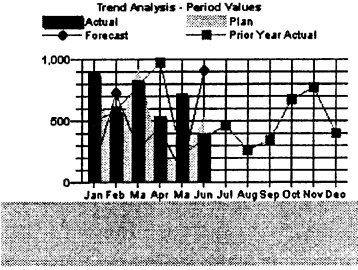
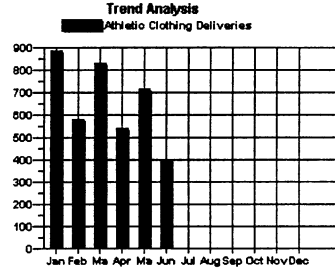
© Copyright Gentia Software, 1999

The *Performance Trend* is the same as the *Version Trend Analysis* page described earlier.

By selecting the *Compare with other measures* button the user will select the third menu option *Actual / Plan Comparisons*.

Actual / Plan Comparisons

1998 Monthly Refresh Data Measure Selection
 Actual World Wide Cosmo Enterprises All Products



	Jan	Feb	Mar	Apr
Athletic Clothing Deliveries	887.00	583.00	833.00	544.00
YTD	887.00	1,470.00	2,303.00	2,847.00



© Copyright Gentia Software, 1999

The *Actual / Plan Comparisons* page is the same as the *Measure Trend Analysis* page described earlier. By selecting a value and choosing the option *Analyze Measure in more Detail*, the user can navigate to the Ad-Hoc analysis page.

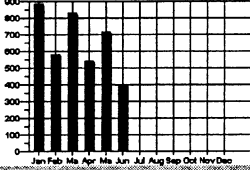
Ad-Hoc Analysis

Database: GPM View	Rows: Location	Columns: Product	Select Options	Refresh Data
Date Start: 1998	Feb	Actual	Athletic Clothing Deliveries	Period
Cosmo Enterprises				

	All Products	Accessories	Clothing	Shoes
World Wide	583.00			583.00
Europe				
South Pacific				
United States	583.00			583.00

Trend Analysis

Athletic Clothing Deliveries



Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Graph Type: 2D Vertical	Dimension Nesting: <input type="checkbox"/> On <input checked="" type="checkbox"/> Off	Data Value Suppression: <input type="checkbox"/> On <input checked="" type="checkbox"/> Off	Dimension Hierarchy: <input type="checkbox"/> On <input checked="" type="checkbox"/> Off	Pivot: Refresh Data
-------------------------	--	---	--	---------------------

GENTIA™

© Copyright Gentia Software, 1999

This page is based on the standard Framework Ad-Hoc page. There is a wide range of analysis options available.

YTD values are not available on this page.

Drill Through

- ◆ Cannot drill through to non-Gentia applications

	C	S	W
Employees			
Revenue per Head			
Profit per Head			
Headcount			
Costs per Head			
Financial			
Sales Credits			
Total Revenue			
Trading Profit			
Project Budgets			
Economic Profit			
Total Costs			
Sales			

RBSC

GENTIA™

© Copyright Gentia Software, 1999

The drill through page (on the web) will have the same name as the drill through name for the client with the word "web" inserted after the three letter application code.

So if the client name for a page is defined as "FWK:XYZ Analysis" it is assumed that the web drill through page name will be "XYZ Web Analysis".

Summary

- ◆ Login
- ◆ Navigate
 - move around the various screens
- ◆ Page layouts
 - what each screen shows
- ◆ Drill through
 - using the RBSC web product to analyze data

GENTIA™

© Copyright Gentia Software, 1999

Chapter 5

Administration and Customisation

GENTIA™

© Copyright Gentia Software, 1999

What we will cover...

- ◆ *.ini* file and architecture
- ◆ Login
- ◆ Administration cycle
- ◆ Schema and Table structures

GENTIA™

© Copyright Gentia Software, 1999

INI file and Architecture

- ◆ *.ini* file connects to the schema files and other data sources
- ◆ Architecture consists of:
 - Relational Schema
 - GentiaDB OLAP Database
 - GPM Administration
 - GPM Analysis Screen

GENTIA™

© Copyright Gentia Software, 1999

What the GPM.ini file should look like.

[GPM]

SqlDataName=Gentia PerformanceImpact

SqlDataSource=GPMSAMP

SqlDataService=GPMSQLSERVICE

SqlDataId=

SqlDataPassword=

[ORACLE]

SqlDataName=Oracle Data Mart

SqlDataSource=ORACLE

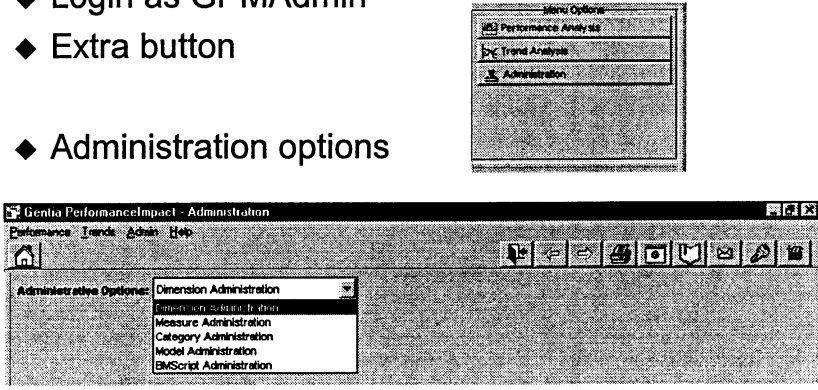
SqlDataService=ORACLESQLSERVICE

SqlDataId=ADMIN

SqlDataPassword=ADMIN

Login

- ◆ Login as GPMAAdmin
- ◆ Extra button
- ◆ Administration options



The image contains two screenshots. The top screenshot shows a 'Menu Options' dialog box with three items: 'Performance Analysis', 'Trend Analysis', and 'Administration'. The bottom screenshot shows the 'Gentia PerformanceImpact - Administration' window. It has a menu bar with 'Performance', 'Trends', 'Admin', and 'Help'. Below the menu bar is a toolbar with various icons. A dropdown menu is open under 'Administrative Options', listing: 'Dimension Administration', 'Pattern with External In-Depth', 'Measure Administration', 'Category Administration', 'Model Administration', and 'BMScript Administration'. At the bottom left is the 'GENTIA' logo and at the bottom right is the copyright notice '© Copyright Gentia Software, 1999'.

An extra option appears when the user is GPMAAdmin. This takes the *Administrator* to a new page. This page shows the *Administrative Options* available. These are:

- Dimension Administration
- Measure Administration
- Category Administration
- Model Administration
- BMScript Administration

Dimension Administration

Name:	DataView	Author Name:	GPAdmin	New
Code:	VIEW	Change Date:	Dec 02 1998 13:22:08	Edit
Structure:	Leaf			Print
Dimension Source Information				
Delimiter:	.	Data Source:	Oracle Data Mart	Open Data
SQL Query:	SELECT DISTINCT VEW_CODE, VEW_DESC, VEW_PARENT, VEW_EXPRESSION FROM GPM_DATAVIEW			Close
				Cancel

GENTIA™

© Copyright Gentia Software, 1999

This page allows the Administrator to add, edit or update dimensions and dimension members to Gentia Business models.

Measure Administration

Name: Athletic Clothing Deliveries	Author Name: GPWAdmin	New
Code: ADEL	Change Date: Oct 08 1998 10:34:39	Edit
Unit: #	App Name:	Roll
Periodicity: Monthly	App Report:	Delete
Definition		Close
Event		Print
Discrepancy		
<input checked="" type="checkbox"/> Exception	System Thresholds	<input checked="" type="checkbox"/> Measure Error is BEC
<input type="checkbox"/> Credit for	If "Plan Var %" > than: 5.00	Approved Code: S00004
<input checked="" type="checkbox"/> Product	If "Plan Var %" < than: -1.00	Operating Code: 000007
RBSIC Information		
Calculation Information		
<input checked="" type="checkbox"/> If-Or Calculation	<input type="checkbox"/> Calculated	<input type="checkbox"/> Mapped
Consolidation: Full	YTD Calc: Sum	<input type="checkbox"/> Reverse Sign of Variance
Expression:		
Formula:		

GENTIA

© Copyright Gentia Software, 1999

This page allows the Administrator to define, edit and update measures. In addition, they can create new measures.

Category Administration

Name:	Employees
Author Name:	GPMAdmin
Change Date:	Oct 06 1998 11:21:20
Code:	CAT_EMP
Measures:	Costs per Head Headcount Funds per Head Revenue per Head

GENTIA™ © Copyright Gentia Software, 1999

This page allows the Administrator to group measures into categories for focused analysis. It is possible to edit, update and create categories.

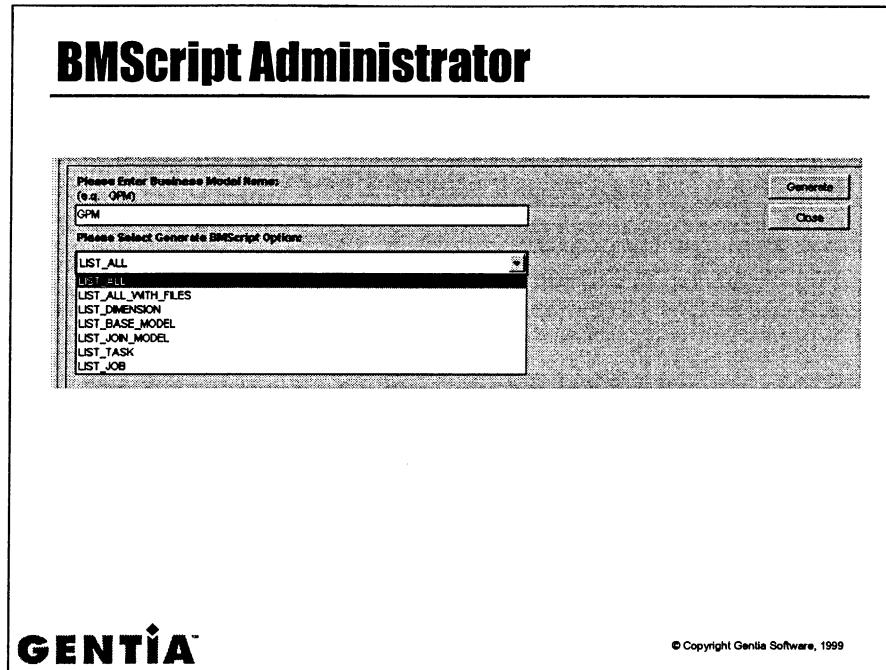
Model Administration

Actions to be performed on MODELS derived from Measure Dimensionality

- Update TIME_ORG
- Update TIME_LOC_PRD
- Update TIME_LOC_ORG_PRD
- Update TIME_LOC

GENTIA™ © Copyright Gentia Software, 1999

This page allows the Administrator to query the measures dimension information and assign them to appropriate base models and join models.



This page allows the Administrator to list and select BMScripts associated with a specific Business Model.

Schema and Table Structures

- ◆ Schema are used to store:
 - dimension definition
 - measure attributes
 - pointers to comment text objects
 - GentiaDB business model status

- ◆ Developed in Microsoft Access but transferable to any relational database.

GENTIA™

© Copyright Gentia Software, 1999

GPM_BM - stores the state of the business model

GPM_CATEGORY - stores category definitions

GPM_COMMENT - stores an index to a text object where the comment is stored

GPM_DATAVIEW - stores DataView dimension member definitions

GPM_DIMENSION - stores dimension definitions

GPM_MEASURE - stores measure definitions

GPM_MEASURE_CATEGORY - stores measure category linkages

GPM_MEASURE_THRESHOLD - stores measure system and user threshold values

GPM_MODEL - stores models available in the business model

GPM_PERIOD - stores period dimension member definitions

GPM_VERSION - stores version dimension member definitions

GPM_YEAR - stores year dimension member definitions

Summary

- ◆ GPM.ini file and schema files
- ◆ Login as Admin
- ◆ Administrator options
 - Dimension
 - Measure
 - Category
 - Model
 - BMScript

GENTIA™

© Copyright Gentia Software, 1999



Gentia - Supported Platforms

Gentia Supported Platforms

- [Gentia 5.0 Supported Client Platforms](#)
- [Gentia 5.0 Supported Server Platforms](#)
- [Renaissance Balanced Scorecard Application](#)
- [Gentia WebSuite](#)
- [Gentia Add-In for Microsoft® Excel](#)
- [SQL Support](#)
- [Answers to common questions](#)

[Download this document](#) in its original Word97 format (26K).

Gentia 5.0 Supported Client Platforms

Client	Details	Minimum Specification		
		Memory (MB)	Processor	Comms
Windows NT™	Workstation 3.51, 4.0	16	Intel 486	TCP
Windows 95 / 98	Windows 95 / 98	16	Intel 486	TCP
Mac(PPC)	MacOS 7.5.3 Revision 2, MacOS 8.0	32	PowerPC	OT/TCP
SUN	Solaris 2.5 & above	16	SunSparc / UltraSparc	TCP

Client Notes
<ul style="list-style-type: none"> • Mac PowerPC Requires Open Transport v1.1.1 or Above • Minimum disk requirements will depend on the size of the Gentia executable and the application(s). The size of the Gentia executable ranges from 3-10 MB depending on the platform. In application terms, a Gentia Object Store (GOS) will be approximately 10 MB and the size of a GentiaDB will be dependent on the amount of data stored.

Gentia 5.0 Supported Server Platforms

Server	Details	Minimum Specification			
		Memory (MB)		Processor	Comms
		STD	GWS		
Windows NT™	Advanced Server 3.51, 4.0	32	64	Intel 486	TCP
DRS6000	DRS/NX 6000 SVR4 Version7MPlus	32	64	Sparc	TCP
HP 9000	HP-UX 10 B.10.01, 11 & above	32	64	PA-RISC 1.1	TCP
RS6000	AIX 4.1.4 & above	32	64	PowerPC 604,601 and POWER 2	TCP
SUN	Solaris 2.5 & above	32	64	SunSparc / UltraSparc	TCP
Windows NT™	Advanced Server 4.0	32	64	Dec Alpha	TCP
Unixware	Unixware 2.0	32	64	Intel 486	TCP
Generic SVR4	(see Server Notes)	32	64	Intel 486	TCP
Pyramid	Data Centre & Nile	32	64	Mips	TCP

Server Notes

- The memory requirements for the Gentia WebSuite (GWS) are different from the standard Gentia server (STD). In addition, GentiaDB may require additional memory and this will be dependent on the size of the application.
- Minimum disk requirements will depend on the size of the Gentia executable and the application(s). The size of the Gentia executable ranges from 3-10 MB depending on the platform. In application terms, a Gentia Object Store (GOS) will be approximately 10 MB and the size of a GentiaDB will dependant on the amount of data stored.
- The Generic SVR4 version of Gentia should operate on any Intel based hardware running a version of Unix which is SVR4 compatible. For example, some NCR and Sequent machines are SVR4 compatible and therefore it is possible to run the Generic SVR4 version of Gentia on this hardware. Please note that in these circumstances Gentia Software will only support the platforms explicitly mentioned in the above document. In the cases where Gentia operates on other platforms due to compatibility between operating systems Gentia Software will only support the core version of Gentia not the specific hardware. Therefore, in the above example we would only support the Generic SVR4 version of Gentia and not the NCR or Sequent platform.
- The Generic SVR4 version is equivalent to Unixware 1.1 and not Unixware 2.0, which

supports multi-threading.

Renaissance Balanced Scorecard Application

The Renaissance Balanced Scorecard application is supported on the following client / server platforms:

Client Platforms	Details
Windows 95™ / 98™	Windows 95 / 98
Windows NT™	Workstation 3.51, 4.0

Server Platforms	Details
Windows NT™	Advanced Server 3.51, 4.0
RS6000	AIX 4.1.4 & above
HP	HP-UX 10, 11 & above
SUN	Solaris 2.5 & above

In addition the following browsers are also supported:

- Netscape Navigator v4.xx
- Internet Explorer v4.xx
- Hot Java Browser v1.1.2

Gentia WebSuite

The Gentia WebSuite is supported on all current Gentia server platforms.

Gentia Server	Details
Windows NT™	Advanced Server 3.51 / 4.0
DRS 6000	DRS/NX 6000 SVR4 V7M Plus Level 7 Increment 6
RS6000	AIX 4.1.4 & above
Dec Alpha	NT 4.0

Pyramid	Data Centre & Nile
SUN	Solaris 2.5 & above
HP	HP-UX 10 B.10.01, 11 & above
Unixware	2.0
Generic SVR4	Intel SVR4

In order to access a Gentia server from a Web browser a number of CGI (Common Gateway Interface) components will need to be resident on the Web Server. Currently, the Gentia Web Server CGI components are supported on the following platforms:

Web Server	Details
Windows NT™	Advanced Server 3.51 / 4.0
RS6000	AIX 4.1.4 & above
HP	HP-UX 10 B.10.01, 11 & above
SUN	Solaris 2.5 & above
Dec Alpha	NT 4.0

The architecture of the Gentia WebSuite effectively means that a Gentia page is processed and built on the Gentia server rather than the Gentia client. Therefore, the memory requirements for the Gentia server, including the Gentia WebSuite, will be greater than a standard Gentia server.

All web browser products supporting HTML 3.2 and Java should be able to access the Gentia WebSuite. The following browser products have been extensively tested:

- Netscape Navigator v3.0, v4.0
- Microsoft Internet Explorer v3.01, v4.0 Minimum version [3.0 (4.70.1215)]
- SUN Javastation

Gentia Add-In for Microsoft® Excel

The Gentia Add-In for Microsoft™ Excel is supported on all current Gentia server platforms excluding Netware NLM.

Gentia Server	Details
Windows NT™	Advanced Server 3.51 / 4.0
DRS 6000	DRS/NX 6000 SVR4 V7MPlus Level 7 Increment 6

RS6000	AIX 4.1.4 & above
Dec Alpha	NT 4.0
Pyramid	Data Centre & Nile
SUN	Solaris 2.5 & above
HP	HP-UX hp-ux10 B.10.01 & above
Unixware	2.0
Generic SVR4	Intel SVR4

The Gentia Add-In for Microsoft© Excel is supported on the following client platforms:

Client	Details
Windows 95™ / 98™	Windows 95 / 98
Windows NT™	Workstation 4.0

The Gentia Add-In for Microsoft© Excel is supported on the following Microsoft Office versions:

Microsoft Office	Details
Office 95	Excel version 7 (Add-In release 2.0d only)
Office 97	Excel version 8

SQL Support

Gentia Software is licenced by Intersolv to distribute a number of ODBC drivers specifically for use with Gentia. Intersolv ODBC drivers are distributed with the Gentia software for the following platforms:

Gentia Client	Gentia Server
Windows 3.1x™	Windows NT
Windows NT™	OS/2 Warp
Windows 95™	HP-UX 10.10, 10.20
OS/2 Warp	Sun Solaris 2.5, 2.6

