

Product Documentation

ER/Studio® Data Architect

New Features Guide

Version 9.5/XE3 Published December 20, 2012 © 2012 Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos, and all other Embarcadero Technologies product or service names are trademarks or registered trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

December 20, 2012

Contents

New Features Summary
General Updates
DBMS Updates 4
New Platform Versions
New Platforms
New Operating Systems5
New DBMS Features
Repository Enhancements
Details of New Features
Partial Layout
Custom Datatypes
Universal Naming Utility Provides Access to Object Editors
Universal Mappings
Creating Universal Mappings through the Compare/Merge Utility
Creating a Universal Mapping using the Universal Mapping Editor
Viewing Universal Mappings17
GENERATED BY DEFAULT for DB2 UDB 9.x
Range-partitioned Universal Tablespaces for DB2 z/OS 9.x and 10.x
ORGANIZE ON for Netezza 6.0
COLLECT STATISTICS for Teradata 12.0 and 13.x
CHECKSUM for Teradata 12.0 and 13.x
Group Support in Repository Security Center

NEW FEATURES SUMMARY

The new features supported by ER/Studio Data Architect 9.5 fall into the following categories.

- General Updates
- DBMS Updates
- <u>Repository Enhancements</u>

GENERAL UPDATES

Graphics Improvements: You can now select multiple objects (entities, views, and shapes) and apply a layout command. All relationships, view relationships or lines that belong to the selected objects will be affected by the layout. You can use existing layout commands from the menu and toolbar. If you want to layout the entire submodel, no objects should be selected. For more information, see <u>Partial Layout</u>.

Custom Datatypes: Using the Datatype Mapping Editor, you can now create your own custom datatypes. For more information, see <u>Custom Datatypes</u>.

Universal Naming Utility: You can now access Object Editors from within this utility. For more information, see <u>Universal Naming Utility Provides Access to Object Editors</u>

Universal Mappings: Mappings you create for Tables, Attributes, Views and Shapes are now automatically matched when comparing and merging models. In addition, mapping you force in the Compare and Merge Utility can be saved in the target model for use in future compare/merge operations. You can see and edit the mappings you create on the Where Used tab of the object. For more information, see <u>Universal Mappings</u>.

DBMS UPDATES

NEW PLATFORM VERSIONS

IBM DB2 z/OS 10.x

Microsoft SQL Server 2012

Netezza 6.0

Teradata 13.x

PostgreSQL 9.x

NEW PLATFORMS

GreenPlum 4.x

NEW OPERATING SYSTEMS Windows 8

NEW DBMS FEATURES

DB2 LUW 9.x: TIMESTAMP datatypes now support generated by default option. For more information, see <u>GENERATED BY DEFAULT for DB2 UDB 9.x</u>.

DB2 z/OS 9.x and 10.x: Range-partitioned Universal Tablespaces. For more information, see <u>Range-partitioned Universal Tablespaces for DB2 z/OS 9.x and 10.x</u>.

Netezza 6.0: Organizing keys now supported. For more information, see <u>ORGANIZE ON</u> for Netezza 6.0.

Oracle 10g and 11g: Capacity planning of tables can now be based on the average row size.

Teradata 12 and 13: ST_GEOMETRY datatype is now supported.

COLLECT STATISTICS and CHECKSUM clauses are now supported. For more information, see <u>COLLECT STATISTICS for Teradata 12.0 and 13.x</u> and <u>CHECKSUM for Teradata 12.0</u> and 13.x

PostgreSQL 8.0 and 9.x: Support has been added for uuid, money, xml, and serial datatypes; uuid is mapped to the UNQUEID logical type; money is mapped to MONEY logical type, xml is mapped to XML logical type; and serial is mapped to SERIAL/INTEGER logical type.

REPOSITORY ENHANCEMENTS

Windows 8: Now supported for ER/Data Architect Repository.

Security Center: Security groups are now supported. For more information, see <u>Group</u> <u>Support in Repository Security Center</u>

DETAILS OF NEW FEATURES

PARTIAL LAYOUT

You can now select multiple objects (entities, views, and shapes) and apply a layout format to that selection only. All relationships, view relationships or lines that belong to the selected objects will be also affected by the layout. This allows you to choose a different layout style to draw attention to specific parts of the model, such as objects recently added or a special relationship that exists between certain entities. If you don't have any objects selected, the layout style chosen will apply to the entire model.

In this example, both Tree and Circular layouts are used.



To change the layout of a portion of model:

1 Drag across the relevant objects to make a broad selection.



- 2 Control-click to add or remove entities, views, shapes, relationships, view relationships, and lines from the selection as required.
- 3 Choose the layout style to apply to the selection from either the **Layout** menu or by clicking the appropriate layout icon on the toolbar, and click **OK** on the warning dialog that appears..

		Contraction of the second s	
	1	Partial Layout Warning Dialog	2
Process of the second of the s	resolution management langer l	ER/Studio Data Architect now supports partial layouts of selected objects. Only the objects currently selected will be affected by the layout. To apply a layout to the full submodel, dick Cancel and deselect all objects in the submodel, then apply the layout again. Otherwise, dick OK to continue with the partial layout of selected objects.	
Caratano Segnore Machinekore Laborkore	Tither Sacce Recal SkaPiane	Product Strengthered Strengther	Cango
Lodita Jaio rovputi Vuotestaa	Crankser Picsreörgie Picsreätse ProducPhos®	Comment of the second s	ati na
	Maarial Color Product Ina Dyla ElizarDopartanea Machill		
	Nortectus		
		Control States States States	
		⇒liodfadban ⇒liodfadban	

Only the objects you selected are laid out using the different style.

CUSTOM DATATYPES

Some platforms have datatypes that are not supported by ER/Studio Data Architect. You can define custom datatypes in the Datatype Mappings editor to use so unsupported datatypes will not be assigned a default datatype, such as CHAR (10) when importing, reverse engineering or merging databases that contain those unsupported datatypes.

The Datatype Mapping Editor has been updated to accommodate this new functionality.

New Custom Datatype	Datatype Mapping Editor Select Mapping: XML Sch Physical Platform: XML Sch Physical Platform: XML S Mappings Logical Datatype BIGINT BINARY BIT CHAR COUNTER DATE DATETIME DATETIME DATETIME DATETIME DATETIME DATETIMN DECIMAL DECIMA	ema (syst Schema Schema Short LIN BIN BIT CH CNT D DT DTO DTO DTO DTO DTO DTO	New Custom Datatype Create a new datatype that can be used in Logical models and D type will be mapped to a physical datatypes for each platform in columns generated from the new logical type will display the corr Enter the logical name of the datatype. This name will appear in Logical Models and Data Dictionary objects. Assign an initial physical mapping or create a new one. The or will appear for physical columns generated on the current pl Schema) using this custom datatype. base64Binary The new custom datatype will be added to the system mapp platforms. The initial physical mapping for this new datatype platforms will be copied from the following existing logical datatype BIGINT	Aata Dictionaries. This new logical this mapping file. Physical responding physical datatype. This short name uniquely identifies the datatype. datatype below latform (XML
---------------------	--	---	---	---

A new option, **Create custom datatypes for unsupported types** has been added to the Reverse Engineering Wizard, which if selected, any datatypes not recognized will be added as new logical and physical types to the selected mappings file.



If the selected mappings file contains new datatypes because you added them or they were added by a previous reverse-engineering process, then those datatype mappings will be used.

NOTE: Width and scale information of unsupported and unmapped datatypes is not retrieved in the reverse-engineering process. If you need this information, create a custom datatype manually through the Datatype Mapping Editor before reverse-engineering.

Usage Example

The PERIOD datatype for Teradata 13.x is not supported natively by ER/Studio. However, you can reverse engineer columns from a Teradata database that uses the PERIOD datatype by checking the option to create a custom datatype.

UNIVERSAL NAMING UTILITY PROVIDES ACCESS TO OBJECT EDITORS

On the **Search Results** tab of the Universal Naming Utility (UNU,) you can double-click on an object and open the editor for that object. In this process, you are asked to save any changes you made in the UNU; the UNU closes; the object is located and selected in Data Model Data Dictionary, or Data Lineage tree; and the object editor opens with the object selected.

- 1 From the Tools menu, select Universal Naming Utility.
- 2 Enter a **Search String** and then click **Search**.
- 3 On the **Search Results** tab, double-click any object and the editor for that object appears.

Universal Naming Utility		
String Specification Search String employee Replacement String	Options ✓ Ignore Case ✓ Whole Word Range Prompted Global (Replace All) 	Apply Close
Search Results Object Scope Model Scope	Create Submodel Rep	ort
Attribute Definition : Contact (store employed Entity : VendorContact Attribute : ContactID Attribute Used Attribute Definition : Contact (Vendor employed Entity : PurchaseOrderHeader	ee) identification number. Foreign	

For example, if you double-click an **Entity**, the **Entity Editor** appears. If you double-click a **Procedure**, the **Procedure SQL Editor** appears.

UNIVERSAL MAPPINGS

Universal Mappings have been implemented to support enterprise-level modeling in large organizations with many systems and overlapping data and processes. They allow you to map objects between models in the same dm1 file, between models in separate dm1 files, and between an ER/Studio Data Architect diagram and a ER/Studio Business Architect workspace in the Repository. These mappings are retained and can be used when comparing and merging. They are automatically applied in the **Compare and Merge Utility** if they exist.

What is a mapping?

A mapping is a link between two objects. There are two types of mappings that serve different purposes.

- **Mappings within a design layer**: This horizontal type of mapping shows reuse within a design layer, such as logical to logical, physical to physical and conceptual to conceptual. The scope of this mapping is limited to mapping objects of the same type, such as entity to entity, table to table, and column to column.
- Mappings across different design layers: This vertical type of mapping shows traceability or where used, upstream and downstream across different design layers, such as: conceptual to logical; logical to physical; and logical and physical.

Why are cross-repository mappings important?

- Impact Analysis: Understanding where common objects are located. Traceability between conceptual and logical models.
- **Scalability**: A mapping system across dm1s and workspaces reduces the need for large models that have everything in them. You can now perform Where Used (ER/Studio DA) or Impact Analysis (ER/Studio BA) that can include information on multiple dm1 or workspace files. Organizing objects into different dm1s or workspaces and then mapping related objects scales much better.
- Performance: Smaller dm1 or workspaces improves performance and manageability.
- Enterprise Modeling: Facilitates an enterprise view of your data. Typically this starts with identifying subject areas, then conceptual entities and then logical/physical models from the conceptual. Once the hierarchy is in place, you can see upstream from logical/physical objects and downstream from conceptual and process models.

The following topics help you understand how Universal Mappings are implemented in ER/Studio:

- Creating Universal Mappings through the Compare/Merge Utility
- Creating a Universal Mapping using the Universal Mapping Editor
- <u>Viewing Universal Mappings</u>

CREATING UNIVERSAL MAPPINGS THROUGH THE COMPARE/MERGE UTILITY

The following illustrates the process of creating and using a **Universal Mapping**:

- **NOTE:** User-Defined Mappings can be used in a similar way when comparing between two models in the same dm1 file. Existing user-defined mappings will now be considered when matching in the compare/merge, and you will have the option to save matches as User-Defined Mappings for intra-diagram comparisons.
- **NOTE:** The precedence for matches in the Compare/Merge Utility is as follows, from high to low precedence: force matches, universal mappings, name matches, attribute sequence number and datatype matches. In other words, if the datatype matches between two attributes but the universal mapping matches one of those attributes to a different attribute, then the second match will be automatically applied. You can use **Force Match** in the **Compare and Merge Utility** to implement a different match.





The **Compare and Merge Utility** automatically matches Person.ADDRESS to Person.Address, but if we want to match Person.ADDRESS to Person.Address1, then we can unmatch these attributes and create a different match.

Compare a	nd Merg	ge Utility - Page 5 of 5	i		0
Current Model:	Advent	ure Works1(Adventure V	Vorks Resolution	Target Model: Adventure Works2(Adventure)	Ν
🗆 🚞 Adver	nture Wo	orks		🗆 🚞 Adventure Works	
🗆 🚞 Ta	bles			🗄 🧰 Tables	
÷ 🚽	ADDD	-00	gnore	🗉 📷 Address	
₽ _	AE	Match Objects	gnore	🕀 💳 Address Type	
₽ _	CC	Unmatch Objects	onore	🕀 💳 Contact	
₽ _	CC	-	gnore	🕀 💳 Contact Type	
₽ _	CC	Expand	gnore	🕀 💳 Country Region	
	STATE	FROVINGE	Ignore	StateProvince	
±-≓	Does N	lot Exist	Ignore	⊡- <mark>=</mark> Address_1	

To illustrate the power of Universal Mapping, we can match two objects and then have the matches saved.

As indicated in the **Compare and Merge Utility**, "Objects may be force-matched by leaving the 1st object select and selecting the target object via right mouse click."



Now that the match is made, we can choose to save the match to use the next time we compare these two models. The matches are saved in both the current and target models.

🛐 Compare and Merge Utility - Page 5 of 5			
Current Model: Adventure Works1(Adventure Works.	Resolution	Target Model: Adventure Works2(A	dventure W
E 💼 Adventure Works		E 📄 Adventure Works	
E Tables		🖃 🧰 Tables	
ADDRESS	Ignore	Address_1	
ADDRESSTYPE	Ignore	🕀 📷 Address Type	
E CONTACT	Ignore	🕀 🚞 Contact	
E CONTACTTYPE	Ignore	🕀 📷 Contact Type	
E COUNTRYREGION	Ignore	🕀 💳 Country Region	
E STATEPROVINCE	Ignore	E StateProvince	
🗈 📷 Does Not Exist	Ignore	🗄 💳 Address	
Objects may be force-matched by leaving the 1st object via right mouse click.	t selected and selecting	g the target object 🔽 Save Ma	tches

You can click the ellipses to see the **Save Matches Options** and make your choices.

lect Matches to Save	Mapping Type
Create mappings for Force-Matched objects so they vill match automatically in future compares.	Universal Mappings can be created between objects across dm1 files.
Save only forced matches	Save as Universal Mappings
Or create mappings for all matched objects. Note: Mappings will be created even for matched objects that are hidden with no differences.	Only matches between entities/tables or attributes/columns within the same file can be saved as User-Defined Mappings.
ve matches for the following object types: Entity/Table Attribute/Column View	

A checkbox for each object type supported, Entity/Table, Attribute/Column, View, and Shapes is listed and all are selected by default. Only objects of the types selected are eligible to have matches saved as mappings.

When comparing models in the same diagram, you can choose between **Save as User-Defined Mappings** or **Save as Universal Mappings**.

When comparing models in different diagrams, **Save as Universal Mappings** is selected. This is the only option available.

If you are comparing two diagrams in the Repository, the option to **Synchronize mapping** changes to Repository immediately upon clicking Finish becomes available.

Click **OK** on the **Save Matches Options** and then click **Finish** to complete the compare/merge.

When you run the **Compare and Merge Utility** again, you will see that the matches you forced and saved are used by default.



CREATING A UNIVERSAL MAPPING USING THE UNIVERSAL MAPPING EDITOR The Universal Mappings Editor provides the following functionality:

- Local Objects You can edit the Universal Mappings of a source object without exiting the Universal Mapping Editor.
- **Browse/Filter/Search**: You can view objects in other diagrams and workspaces that can be mapped to the currently selected object. Because the number of objects can potentially be very large, you can narrow down the list by using the filters available and sorting the related objects.

- **Existing Mappings**: You can see which objects are already mapped, as well as the description and any other data associated with the mapping.
- 1 In the diagram window, right-click the object you want to map or change the mapping for and select **Universal Mappings**.

xpand Diagrams and Workspaces in the Repository to	Select an	object fr	rom the local diagram	n in order to view and ed	it the		 Vertical M 	lappings	T
now available models, then select models to browse.	monoto a	r for that	t ablant				Filter By	Objec	
niers available for maboling will be shown in the drid on		Logical Main M	Indal			*	E Entity/Ta	able	
empjasonnut	1	Entil	Hor			-	E Attribute	Column	ļ
Sample Models	Î		OVED			=	P View	(Cordinin	1
Adventure Works.DM1			ROKER DIM				☑ Shape		1
Adventure WorkS_JH.DM1		A B	ROKER DIM 1				Business	Archite	
RO Emergency Admissions dm1		B B	ROKER DIM 1 1				Business	Entity	1
GIMDB DM1		de d	ient			-	Business	Attrib	
Logical	Sort and	filter the	list below to find rel	ated objects to man to th	e selecter	Llocal	object		i
GIM_DW	Mapped	Type (Object Name	(Parent) Object Name	Model		Diagram/W	Vorkspace	5
-GA GIM_OLTP		Filter:							ŀ
Northwind.dm1	0	B	roker	Broker	Logical		GIMDB.DM	1	1
Galacian de 1	2	B	roker Dim	Broker Dim	Logical		GIMDB.DM	1	
	0		lient	Client	Logical		GIMDB.DM	1	
	0	C	lient Facts	Client Facts	Logical		GIMDB.DM	1	
			lient Transaction	Client Transaction	Logical		GIMDB.DM	1	
		😂 D	VimensionalView	DimensionalView	Logical		GIMDB.DM	1	
		I	nvestment	Investment	Logical		GIMDB.DM	1	
		📑 I	nvestment Dim	Investment Dim	Logical		GIMDB.DM	1	
		📄 I	nvestment Type	Investment Type	Logical		GIMDB.DM	1	
			Office Location	Office Location	Logical		GIMDB.DM	1	
		📑 R	legion Dimension	Region Dimension	Logical		GIMDB.DM	1	
		🌔 S	hape1	Shape1	Logical		GIMDB.DM	1	
		🌓 S	hape2	Shape2	Logical		GIMDB.DM	1	
		🌓 S	hape3	Shape3	Logical		GIMDB.DM	1	
		🌓 S	hape4	Shape4	Logical		GIMDB.DM	1	
	1.5	• •	hanos	Chanos	Looirol		CIMDB DM		

- 2 In the Universal Mapping Editor, in the center of the window, as indicated onscreen, **Select** an object from the local diagram in order to view and edit the mappings for that object.
- 3 On the far left of the screen select a model in the tree. Once you have selected the model, the data for it is retrieved from the Repository.

The list of mappable objects that displays on the bottom right of the editor is generated from your selection in the Repository tree. If you select a diagram node, then all objects in its models will be included in the mappable object list. If you select a Project, then all objects in diagrams in that project will be included.

- 4 On the far right of the editor, select the filters to use in order to see the objects of those type you can map to.
 - **NOTE:** You can map ER/Studio DA entities/tables, attributes/columns, views, shapes and ER/Studio Business Architect entities, attributes, and subject areas to each other, including mapping ER/Studio DA objects to ER/Studio BA objects.
- 5 At the bottom of the editor, click a column name to reorder the list of related objects to find the object to map to the local object.
- 6 Deselect or select the target object's **Mapped** checkbox to remove or add a mapping.
- 7 Click **OK** to complete the change.

8 Save your diagram and in the case of Repository diagrams, check in the affected object to save the mapping change to the Repository.

VIEWING UNIVERSAL MAPPINGS

Universal Mappings are displayed in the object's **Where Used** tab.

Entity Editor	ROKER DIM 1	Table Name:		4 1	Owner:	GIMD
Logical Only			[bronch_bi			OIT ID
Attributes	Keys	Relationsh	lips	Definition	1	N
Compare Options	Data Lineag	e !	Security Informat	ion	Attachme	nt Bind
Where Used	Constraints	Dependencies	Pe	rmissions	Namin	g Stan
GIM_DW (Micros)	oft SQL Server) ROKER_DIM le) gs gs lent Management					
				<u>о</u> к	X Cancel	3

You can also see the mappings of an object by right-clicking the object and selecting **Where Used**.

Where Used	×
Show only submodels that use this object Show only submodel stat use this object Content of the state of th	

And you can see the mappings of an object in the **Universal Mappings Editor**.

xpand Diagrams and Workspaces in the Repository to	Select an	objec	t from the local diagram	m in order to view and edit	the	 Vertical Mappings 	
how available models, then select models to browse.	chille	I a minu	hat alstaat			Filter By Objec	
nierts availanie for manning will be snown in the drig on		Logica	n Model		^	Data Architect D	
ethojasonnul			ntition		-	E Attribute/Column	
Sample Models	1 1		PROVER		-	View	t I
Adventure Works.DM1			PROKER DIM			☑ Shape	=
Auvenure works_JR.DM1		T.	BROKER DIM 1			Business Archite	
B Emergency Admissions dm1		J.	BROKER DIM 1 1			Business Entity	4
CIMDR DM1		J.	Client		-	Business Attrib	
	Cort and	filter	the list below to find rel	lated objects to man to the	colocted los	n	-
GIM_DW	Mannad	Tumo	Object Name	(Darent) Object Name	Model	Diagram Workspace	
GIM_OLTP	Mapped	Filter	Tobject Name	[(Parent) Object Name	Model	Diagram/ workspace	1
Northwind.dm1	L	ritter	Broker	Broker	Logical	GIMDB DM1	1
B OrangeMart.dm1		-	Broker Dim	Broker Dim	Logical	GIMDB.DM1	
B Orders.dm1		- 2	Client	Client	Logical	GIMDB.DM1	
	0	- 2	Client Facts	Client Facts	Logical	GIMDB.DM1	
	0		Client Transaction	Client Transaction	Logical	GIMDB.DM1	
		-	DimensionalView	DimensionalView	Logical	GIMDB.DM1	1
	0		Investment	Investment	Logical	GIMDB.DM1	
	0	B	Investment Dim	Investment Dim	Logical	GIMDB.DM1	
			Investment Type	Investment Type	Logical	GIMDB.DM1	
	0		Office Location	Office Location	Logical	GIMDB.DM1	
			Region Dimension	Region Dimension	Logical	GIMDB.DM1	
		- 6	Shape1	Shape1	Logical	GIMDB.DM1	
		- 🏠	Shape2	Shape2	Logical	GIMDB.DM1	
		- 🏠	Shape3	Shape3	Logical	GIMDB.DM1	
		- 🏠	Shape4	Shape4	Logical	GIMDB.DM1	١.,
	-		Chanas	Chanas	Logical	CIMDD DM1	

GENERATED BY DEFAULT FOR DB2 UDB 9.X

A generate **By Default** option is now available for a **TIMESTAMP** datatype if the column does not allow nulls (**Allow Nulls?** is set to **No**), and you have selected **Generated Column**.

Columna Dock Generate Permosons Press, A Foots 2, Homing Standards Compare Options Data Lineage Security Defonition: Attachment Bindings Storage Discussions Dock Lineage Security Defonition: Capacity Marring Columns DOC Indexes Processing Constraints Dependencies Capacity Marring Columns DOC Indexes Processing Constraints Defonition Note Where Used Columns DOC Indexes Processing Constraints Dopendancies Capacity Marring Column Domain INTEGER Not NULL Column Columna Domain INTEGER Not NULL Column Column Column Columna Domain INTEGER Not Null Column Column Column Columna Domain INTEGER Not Null Column Column Column Column Columna Domain INTEGER Not Null Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column Column		×	_ 0						le Editor
Permissions PresQL & PostSQL Nerring Standards Compare Options Data Lineage Security Information Capacity Rearing Capacity Rearing Devendencies Dependencies Capacity Rearing Capacity Rearing Dot Indexes Poregot Keys Definition Note Where Used Rearing Dot NULL Indexes Poregot Keys Definition Note Poregot Dot NULL Indexes Poregot Related Dot NULL Indexes Poregot Keys Dot NULL Indexes Poregot Related Dot NULL Protect Dot		-	: HumanResou	Owner:	Employee	Table Name:	ienerate T	Employee	Entity Name ysical Only
Storage Diversions Database Constraints Dependencies Capacity Rearing Columns DOL Indexes Foregrin Keys Definition Note Where Used 1 Storage Exclusion Domain Distance Domain Note Where Used 1 Storage Exclusion Domain Note Where Used Domain 1 Storage Exclusion Domain Note Domain Note 1 Storage Exclusion Domain Note Domain Note 2 Exclusion Undexes Foregrin Keys Definition Note Note 3 Definition Note Workstance Note Note Note 3 Definition Note Note Note Note Note Note 4 Login Contractions Foregrin Keys Note Note Note Note Note 5 Manadata Generate Generation Storage Note Note Note Note Note		gs	Attachment Bindir	curity Information A	Data Lineage Ser	ompare Options	aming Standards Com	8.PostSQL Na	ssions PreSQL
1 Column Domain Note + Under Hiller 1 Engloyee1D National DNarriser WARSARPHIC(15) MOT NULL 2 Destat1D WARSARPHIC(15) MOT NULL 4 LegrilD WARSARPHIC(15) MOT NULL 5 MenggeriD WARSARPHIC(15) MOT NULL 6 Tele WARSARPHIC(256) NOT NULL 5 MenggeriD WARSARPHIC(256) NOT NULL 6 Tele WARSARPHIC(256) NOT NULL 7 Brithisto Martalistatue MARLES 8 Brithisto MARLES MARLES 9 Bender Cancel Physical Only 9 Bender Operation Relearner: Physical Only 9 Bender Operation Relearner: Physical Only 9 Bender Default Column Relearner: Physical Only <			Capacity Planning Where Used	Dependencies C Note	Constraints	Partitions Enceim Keyr	stribute Columns	sions Dis	rage Dimer
Image: Decision Downin In the Decision Downin In the Decision Image: Decision Image: Decision Wassinger Downin Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision Image: Decision I		a li li	Nole	Datatune	omain		underved 1	Column	1
Z National DNumber VARGRAPHIQ15) NOT NULL D Logari0 WARGRAPHIQ256) NOT NULL S Manageri0 WARGRAPHIQ256) NULL Sender Elsenation GRAPHIQ200 NOT NULL Sender Octoberin Hide Key Abriduer Provide Column Releases Managerid Logard Rolenanes Weise Managerid Selection Releases Fribate Default Column Relenanes Matalype Default Rule/Constraint Definition Notes		ál l	IDENTITY	TEGER	IN			COMMIT	EmployeeID
3 Cardiadio INTEGER NOT NULL 4 LegrinD VARGRAPHIC(256) NOT NULL 5 Tate VARGRAPHIC(50) NOT NULL 7 Enhibition NUT NULL INTEGER 8 MensgeridD INTEGER NULL 9 MensgeridD INTEGER NULL 7 Enhibition NULL NOT NULL 7 Enhibition NOT NULL INTEGER 9 MattaBatua GRAPHIC(1) NOT NULL 9 Save Dekte Ide Version 9 Save Dekte Ide Provide 9 Add Save Dekte Ide 9 Add Save Dekte Ide 9 Add Desce Ide Provide 9 Add Denne Provide Provide Ide 9 Default Reference Add to Primery Key? Edit Secondarial Sec Provide 9 Default Rule/Constraint Definition Notes Where Used Ref			NOT NULL	RGRAPHIC(15)	W/			umber	National DN
LegenD VARGRAPHIC(25) NOT NULL INTEGER NULL NULL Title VARGRAPHIC(30) NOT NULL Title VARGRAPHIC(30) NOT NULL MatalSatas GRAPHIC(1) NOT NULL MatalSatas GRAPHIC(1) NOT NULL MatalSatas GRAPHIC(1) NOT NULL Sender GRAPHIC(1) NOT NULL MatalSatas GRAPHIC(1) NOT NULL Sender Grade GRAPHIC(1) NOT NULL MatalSatas GRAPHIC(1) NOT NULL File MatalSatas Grade Grade File Sender Cooperation Hele MatalSatas Sender Cooperation Hele File MatalSatas Cooperation Hele File Sender Cooperation File File Sender Cooperation File File For Bit Data Column Seed: For Bit Data Mittestamp Generated Column Generated Column Generated Column Allow Nulls ? No		F	NOT NULL	TEGER	IN				ContactID
Image: Managerio Introduce NULL Tale NARGRAPHICSO NOT NULL Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio Image: Maragerio			NOT NULL	RGRAPHIC(256)	VA				LoginID
Ite I			NULL	TEGER	IN				Nanagari D
Index Aver Aver Aver Aver Aver Aver Aver Aver			NOT NULL	RGRAPHIC(50)	VA				Ttle
Immendiatenzation Indiring and indirindiring and indirindity and indiring and indiring and indiring an			NOT NULL	ABUICO	61				BithLate Made Date
Add Save Delete Up Down C DK Cancel Help Add Save Delete Up Down C DK Cancel Help Add Save Delete Up Down C DK Cancel Help Add Save Delete Up Oracle Domain Help Key Athlaste Add to Rame: Logical Rolename: Save Save Default Column Rolename Add to Rame Rolename Save Default Column Rolename: Add to Rame Rolename Add to Rame Rolename Add to Rame Rolename Save Default Column Rolename: Add to Rame Rolename Add to Rame Rolename Reference Values Naming Standards C Image: Properties atatype Default Rule/Constraint Definition Notes Where Used Reference TIMESTAMP Identity Column Seed: Identity Column Not Logged C Width: Scale: Increment: Identity Column Generated Column Allow Nulls ? I Image: No Image: No Image: No </td <th></th> <th>-</th> <td>NOT NULL</td> <td>APHIC(1)</td> <td>64</td> <td></td> <td></td> <td>8</td> <td>Fandar Satu</td>		-	NOT NULL	APHIC(1)	64			8	Fandar Satu
Add Sage Dekte Up Down OK Cancel Help nain Name: CNE Image: CNE Image: CNE Oracle Domain Hide Key Attribute indue Name: Logical Rolename: Physical Only Synthemate Column Rolename Physical Only indue Name: Default Column Rolename: Physical Colum Reference Values Naming Standards CI atatype Default Rule/Constraint Definition Notes Where Used Reference Values Naming Standards CI Image: Column Rolename atatype Default Rule/Constraint Definition Notes Where Used Reference Datatype Default Rule/Constraint Definition Notes Where Used Reference IMESTAMP Identity Column Seed: Image: Column Reference Image: Column Reference Allow Nulls ? O O Image: Column Generated Column Generation Expres 1 O O Image: Column Generation Expres Image: Column Generation Expres			AUT HOLE	(i) (ii) (ii)	la.				Creations
TIMESTAMP Identity Property Datatype Properties Width: Scale: Identity Column Seed: I Not Logged C Allow Nulls ? Increment: Generated Column Yes No No Seed:	nce Values Nar	▶. Referen	ndards ಧ[/here Used	ranjes Calumn Rolenam ocisel Rolename o Primary Key? onsign Key Datatype alues Naming Stan Notes Wł	IF Synth IF Add t IF ER F Jsed Reference V Definition	Notes Where U	Dofault Column Rolen	Rule/Constra	ate Column Name: ate type Default atatype
TIMESTAMP Identity Column Width: Scale: 0 0 Allow Nulls ? Increment: Yes No		Properties -	Datatype	erty	Identity Prop	_ ¹			
Not Logged Width: Scale: 0 0 Allow Nulls ? 1 C Yes No		t Data	E For B	Column	Identity				
Width: Scale: I 0 0 Increment: Allow Nulls ? 1 C Yes No	Compact	baaed 🗖 (Not L			▼		MP	ITIMESTA
Width: Scale: 0 0 Allow Nulls ? C Yes Image: No					Seed:				
0 0 1 Increment: Generated Column Allow Nulls ? 1 ✓ Generation Exprese 1 1							cale:	S	Width:
Allow Nulls ? C Yes © No		ated Caluma	E Com		1	_			0
Allow Nulls ? C Yes No	III IV By Defau	ated Column	Je Gene				0		0
Allow Nulls ? C Yes No 	s un:	ation Expres	Gener		Increment:			-	
C Yes © No								s ?	Allow Null
					1			No	C Vec
									e ies

The following is an example of the SQL produced for a column when these options are selected

HireDate TIMESTAMP NOT NULL GENERATED BY DEFAULT FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP.

Range-partitioned Universal Tablespaces for DB2 z/OS 9.x and 10.x $\,$

The Tablespace Editor now for IBM DB2 z/OS (OS/390) 9.x and 10.x lets you specify a new type of tablespace, **Universal (Range-partitioned)**.

A range-partitioned tablespace is segmented and based on partitioning ranges. It contains a single table and you can create any type of index on tables in the range-partitioned tablespace.

The following describes the minimal steps necessary to create a universal range-partitioned tablespace.

- 1 In the Data Model Explorer tree, for a DB/2 OS/390 9.x or 10.x physical model, right-click the Tablespace object and select **New Tablespace**.
- 2 Enter a name for the tablespace and then select **Universal (Range-partitioned)**.
 - **TIP:** You can also specify the type of data to store in the tablespace on the **Name & Type** tab of the **Tablespace Editor**.

DB2 OS/390 Tablespace	te Wizard - Page 1 of 7	×
	A tablespace consists of one or more data sets on your server, which are may by the user or by DB2 itself. You can place one or more tables on a tablespace?	anaged ce.
	 Choose the type of data to store in the tablespace : Non partitioned Partitioned (all types of table objects.) Segmented LOB (long text, audio, video, or images) Universal (Range-partitioned) 	
	Select the database that contains the tablespace.	Finish
X Cancel ? Hel	p Back Ne <u>x</u> t 🤛	Finish

- 3 Select other options you want on page 1 and page 2.
- 4 On page 3 of the Tablespace Wizard, specify the **Number of Partitions**.
- 5 Select a value in the **Segment Size** list.

DB2 OS/390 Tablespace Wizard - Page 3 of 7	×
Select the following options for this tablespace : GBP Cache : GBP Cache : GBP Cache : GBP Cache : Sizing Number of Partitions: Segment Size Segment Size: 12 Max rows per page 125 (default) Free Block FREEPAGE: O PCTFREE: 5	
X Cancel ? Help	🗱 <u>F</u> inish

- 6 Specify the **Max rows per page** or accept the default of 255.
- 7 Click **Finish**.

The following is an example of the DDL produced when creating a universal range-partitioned tablespace.

CREATE TABLESPACE URPT

NUMPARTS 5 SEGSIZE 12 MAXROWS 125

:

ORGANIZE ON FOR NETEZZA 6.0

Organizing keys are now supported for the Netezza platform. The Table editor has been expanded by adding a new Organization tab for the Netezza 6.0 platform.

Physical Only	Do Not Generate	Table	Name: Englands		Uwner: HumanResou
Columns	DOL	Indexes	Foreign Keys	Defi	ition Note
Naming Standards	Compare Opt	ons Data Li	eage Securi	ty Information	Attachment Bindings
Where Used	Distribution	Organizat	ion Capac	ity Planning	PreSQL & PostSQL
Organizing Keys – Available Columns: EmployeeID NationalIDNumber ContactID LoginID	Sek	ected Columns:	Up Down		

On the **Organization** tab of the **Table Editor** for the Netezza 6.0 platform, you can select up to four columns of a table on which to base the organization of the table. Tables with columns specified in the ORGANIZE ON clause are called Clustered Base Tables (CBT). Organizing the table records in this way causes Netezza to save the selected table records in the same or nearby extents. This can enhance query performance and disk I/O.

The following is an example of the SQL produced by specifying organizing keys:

ORGANIZE ON (EmployeeID, ManagerID)

Using ORGANIZE ON Clause

CBTs are most often used for large fact or event tables with many millions or billions of rows. Scanning records of such large tables requires full disk scans to gather the relevant records. Organizing tables to correspond with your queries can take advantage of zone maps to improve performance.



- Available Columns: Displays all the columns available to add to the organization key. Select the column you want to add to the organizing keys and move it to the Selected Columns box. Use the left and right arrows to move columns to and from the Selected Columns box. You can select a maximum of four columns to use in the organizing key.
- Selected Columns: Displays the columns that make up the organizing key.
- **Up/Down** buttons: Let you reorder the columns in the organizing key. The column order can affect the access speed. The most frequently accessed columns should be at the top of the Selected Columns list.

COLLECT STATISTICS FOR TERADATA 12.0 AND 13.X

The Table and Index editors have been updated to accommodate **Collect Statistics** options for the Teradata platform.

The following shows the new **Collect Statistics** and **Using Sample** options on the **Datatype** tab of the **Table Editor**.

Table Ed	ditor							1111			х		
Er	ntity Name:	Employee		Tab	le Name:	Empl	loyee		Owner:	HumanResou	-		
Physical	I Only	Do Not G	enerate								_		
Namin	o Standards	Comos	are Options	Data	Lineage	1	Security In	formation	Atta	chment Bindings			
Where	Used	Storage	Constrai	nts	Dependenci	es	Capa	city Planning	Pr	eSQL & PostSQL			
Colu	umns	DDL	1	ndexes	- Fo	oreign	Keys	Defini	tion	Note			
		Column	1		D	omai	n	Dataty	pe	Nulls	*		
9	Gender							CHAR(1)		NOT NULL			
10	HireDate							DATE		NOT NULL			
11	SalariedRag							Flag		NOT NULL	- 1		
12	VacationHour	8						SMALLINT		NOT NULL			
14	SickLeave Ho Overset Dee	urs						SMALLIN I		NOT NULL			
15	current-tag mwa.id							INTEGER		NOT NULL	- =		
16	ModifiedDate							DATE		NOT NULL			
17	EncryptedNat	ionalIDNumber	r					BLOB(128M)		NULL	-		
Adv	a 🧷 o		Delete	A lh		Dow	. 🖌	OK 3	Conne	о 19 на			
1	· / ·		<u>-</u>	~ *	_					.) [• ··~	Ψ.		
Domain Nam	ne:						_						
[NONE]				-	Create Do	main	E Hr	e Key Attribute					
Attribute Na	ame:		Logical Roli	ename:		_	E Ph	ysical Only					
SickLeaveh	Hours						- I	h Logical Rolena	n Rolenan me	10			
Default Colu	umn Name:		Default Col	umn Rolenam	ie:	_	E Ad	d to Primary Key	12				
SickLeave	Hours						🗖 Eđ	t Foreign Key Di	atatype				
Datatype	e Default	Rule/Constrai	int Defini	tion Notes	Where U	sed	Reference	e Values Na	aming Star	dards Q	Þ		
	Dataty	/pe Def	ault I	Rule/Con	straint	0	Definitio	n Notes	Whe	ere Used	Re	eference Values Na	iming St
SMALLIN	0					Tele-	titu Doo	a sector			_		
Width:						lden	iuty Pro	perty	For	nat:			Title:
0							Identity	Column	Con	-	i-		
Alow Nul	SMALL	INT			<u> </u>	_			Con	ipi ess:			Mor
C Yes						See	d:						
	Width		Scale:			1				· · · ·			
	0		0			11							
	1.		1-			Ind	crement						
	Allow I	Nulls ?				_							
	C V-	G No				1							
	· res	5 VE 140								Collect Sta	tistic	:s I⊻ Using S	ample

The following shows the new **Collect Statistics** and **Using Sample** options on the **Options** tab of the **Index Editor**.

🚯 Index Editor	
Properties Index Name: PK_Employee_EmployeeID ✓ Set as PK Index ✓ Physical Only	V OK Cancel
Columns Options Compare Options Attachment Bindings Index Option: Partitioning Expression More Image: Primary More More Image: Collect Statistics Image: Using Sample More	

The following describes the purpose of these options and their effect on the SQL generated for the physical model:

Collect Statistics: (COLLECT STATISTICS) Collects demographic data for one or more columns of a table, computes a statistical profile of the collected data, and stores the profile in the data dictionary. Collecting full statistics involves scanning the base table, and sorting to determine the number of occurrences for each unique value. The Teradata Optimizer uses this profiling data when it generates its table access and join plans. Full statistics is generally recommended for relevant columns and indexes on tables with less than 100 rows per Access Module Processor (AMP).

This is an example of the SQL generated when **Collect Statistics** has been selected in the **Table Editor** for a column in the table.

COLLECT STATISTICS ON table_1 COLUMN column_1;

This is an example of the SQL generated when **Collect Statistics** has been selected in the **Index Editor** for an index in the table.

COLLECT STATISTICS ON table_1 INDEX unique_1;

Using Sample Collecting sampled full-table statistics is generally faster than collecting full table statistics, but may not produce the best table access and join plans. When you specify

this option, Teradata will determine the appropriate percentage to the data to scan. The best choices for USING SAMPLE are columns or indexes that are unique or 95% unique.

This is an example of the SQL generated when **Collect Statistics** and **Using Sample** have been selected in the **Tale Editor** for a column in the table.

COLLECT STATISTICS USING SAMPLE ON table 1 COLUMN column 1;

This is an example of the SQL generated when **Collect Statistics** and **Using Sample** have been selected in the **Index Editor** for an index in the table.

COLLECT STATISTICS USING SAMPLE ON table_1 INDEX unique_1;

CHECKSUM FOR TERADATA 12.0 AND 13.X

The **Table Editor** has been changed to accommodate the new **Checksum** options for the Teradata platform.

The following shows the new **Checksum** and **Medium** options on the **Storage** tab of the **Table Editor**.

Table Editor							- • •
Entity Name:	SAMPLE_TABLE4	ю	Table Name:	SAMPLE_TABLE40		Owner:	B2OWN99 👻
Physical Only	Do Not Gener	ate					
Columns	DDL	Indexe	s For	eign Keys	Definiti	on	Note
Naming Standards	Compare C	Options	Data Lineage	Security Info	rmation	Attachm	ent Bindings
Where Used	Storage	Constraints	Dependencies	s Capacit	ty Planning	PreSC	QL & PostSQL
Journal Table		Dupl	icate Row Control	(default)	•		
Before Journal	(default) 💌	Chee	cksum		-		
After Journal	(default) 👻	E F	Fallback Protection	 All	^		
Freespace ((default) 💌 P	ercent		DEFAULT HIGH	=		
Datablock Size	(default) 💌 K	в		MEDIUM LOW	-		
					ок 🗙	Cancel	💡 Help 🏼

Checksum (CHECKSUM) : Detects and logs disk I/O errors. The choices available determine the percentage of data used to compute the checksum for the table. This is an example of the SQL generated with **Checksum > Medium** has been selected on the **Storage** tab of the **Table Editor**.

Checksum = Medium

GROUP SUPPORT IN REPOSITORY SECURITY CENTER

To make it easier to handle permissions, group support has been added to the ER/Studio Repository Security Center. Repository administrators can assign roles to the group at repository, project and diagram levels, then simply add or remove users from the group to apply the settings to the users quickly and easily. The Repository administrator can either manually create a group and assign users to it or use and LDAP/Active Directory group and add users to the Repository to a group based on LDAP/Active Directory settings.

On the **Repository Security** tab, the list of **Available Users** displays **Groups** first and then individual **Users**.

ER/Studio Repository Security Center		_ • •
Welcome To the Repository Security Centerl Use the tabbed interface to create and mana they have to Repository Projects, Diagrams a Repository Security Manage Users Mai To create Repository Object, User and Rol Available User then drag and drop the user	age Groups, Users, Roles, and the assignments and Diagram objects. nage Groups Manage Roles e associations. highlight a Repository Object (e.g rto an Available Role the user should perform ag	g. a diagram or Submodel). select an jainst the Object.
Repository Object	Available Users Groups Data Architect Users Jason	Available Roles Super User Admin Data Architect Jacquie Data Architect Supervisor
	🔺 Арр	ly 🗸 QK 🗙 Cancel 💡 Help //

A new **Manage Groups** tab has been added to the **ER/Studio Repository Security Center** where you can add, edit, and delete groups.

Velcome To the Re	epository Securi	ty Center!	Roles, and the a	ssianments		
hey have to Repositor	ry Projects, Diagra	ms and Diagram object	s.	and the second		
Repository Security	Manage Users	Manage Groups	Manage Roles	1		1
Group		Group Description	1			
🔆 Data Archit	ect	Data Architect				
Group Managemen	nt					
-Group Managemen	Edit	Delete				
Group Managemen	Edit	Delete				
Group Managemen	Edit	Delete				

Create, Update, and Delete Group

- 1 Choose **Repository > Security > Security Center**.
- 2 Click the Manage Groups tab and select New.

🙎 Edit Reposi	tory Group	
Name: Description:	Directory Service Group Check Name	
- Members -	Users Users In Group	*
Admin Jacquie		
Creat	e Repository Users for all members of this Directory Ser	vice Group
	OK Cancel Help	

3 Provide a name for the group and optionally, a description of the group.

4 If you want to create a group based on an LDAP/Active Directory group, click **Directory Service Group** and another option becomes available at the bottom of the dialog, **Create Repository Users for all members of this Directory Service Group**. If you select this option, then repository users in this group are automatically added to the list of Repository users. If you do not select this option, then you will need to create the users manually and as you add LDAP/Active Directory Service users, they will be added to the Repository groups that correspond to the LDAP/Active Directory Service groups that they are members of.

Directory Service	Group Ch	eck Name	
escription:			
			*
			_
Members			
Available Users	Us	ers In Group	
Admin embarcadero\Build1	- r		
embarcadero\HenryC embarcadero\iacquelinef			
embarcadero\jasonh embarcadero\JimB			
embarcadero\Oliverf embarcadero\SherwinA			
embarcadero\XuhuaL			
	••		
Create Depository Licers for	all members of this [irectory Service Grou	þ

If you want to manually populate the group, do not select **Directory Service Group**, from the list of **Available Users**, select the group members by Ctrl-clicking each user, or click the double right-facing arrow to select all the users. You can also remove group members by clicking the left-facing arrow.

Notes

- **Delete a group**: On the **Manage Groups** tab, you can delete a group by selecting the group and clicking **Delete**. Deleting a group does not delete the group members but the group members will no longer have the permissions afforded to them by the group membership.
- Edit a group: On the Manage Groups tab, you can edit the group name and list of group members by selecting the group, clicking Edit, and making your changes on the Edit Repository Group dialog.

Assigning a Role to a Group

Assigning a role to a group grants the group all the permissions selected in the role.

- 1 Choose **Repository > Security > Security Center**.
- 2 Select a Repository item you want to assign permissions to.
- 3 If the group is not currently assigned a role, on the **Repository Security** tab, click the group name in the **Available Users** column and drag it onto a role name in the **Available Roles** column.

If the group was already assigned a role, on the **Repository Security** tab, click the group name the **Available Roles** column and drag it onto another role name in the **Available Roles** column.

- 4 Click **Apply** and then click **OK** to exit the **Security Center**.
- 5 Notify users in the group that they must log out and then log in again to receive the security updates.
 - **NOTE:** If a user has been assigned the No Access role for an object, but that user is a member of a group with permissions for the object, then the user will have all permissions assigned to the group. All permissions available to a user through direct assignment to roles are added to the permissions available to any groups the user is a member of. Permissions are additive, so the user gets all the permissions assigned to the user whether they were assigned through a group or directly to the user.