CA ERwin® Data Modeler

Implementation Guide

Release 9.6.0

technologies

CA Technologies Product References

This document references the following CA Technologies products:

- CA ERwin® Data Modeler Standard Edition
- CA ERwin® Data Modeler Workgroup Edition
- CA ERwin® Data Modeler Navigator Edition

Contact CA Technologies

Understanding your Support

Review support maintenance programs and offerings.

Registering for Support

Access the CA Support online registration site to register for product support.

Accessing Technical Support

For your convenience, CA Technologies provides easy access to "One Stop" support for all editions of CA ERwin Data Modeler, and includes the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

For information about other Home Office, Small Business, and Enterprise CA Technologies products, visit http://ca.com/support.

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is also available on the CA Support website, found at http://ca.com/docs.

CA ERwin Data Modeler News and Events

Visit www.erwin.com to get up-to-date news, announcements, and events. View video demos and read up on customer success stories and articles by industry experts.

Documentation Changes

The following documentation updates have been made since the last release (r9.0) of this documentation:

- Table of Supported Bridges (see page 75)—Updated with the list of bridges supported for this release.
- Applying Naming Standards (see page 52)—Updated the topic to clarify the function when naming standards are imported from a previous release.

The following documentation updates have been made since the last release (r9.0) of this documentation:

- Table of Supported Bridges—Updated with the list of bridges supported for this release.

The following documentation updates have been made since the last release (r8.2) of this documentation:

- Create a Model—Updated this topic per the new design changes in the New Model dialog. It explains different procedures that are available in CA ERwin Data Modeler Version 9 to create a model.
- Apply Naming Standards (see page 52)—Updated this topic per the design changes and application of Naming Standards.
- Apply Data Type Standards (see page 67)—Updated this topic per the design changes and application of Data Type Standards.
- Table of Supported Bridges—Removed MIR XMI format export and import bridges.

Contents

Chapter 6: Creating and Using Domains 45

Chapter 7: Applying Naming and Datatype Standards 51

Appendix A: Supported Metadata Integration Bridges 75

Index 81

Chapter 1: Overview

This section contains the following topics:

Database Design Overview

CA ERwin Data Modeler (CA ERwin DM) provides the tools to design and implement databases for transactional business, E-commerce, and data warehousing applications. You can create and maintain graphical models that represent databases, data warehouses, and enterprise data models. You can also define, manage, and implement corporate data requirements on several databases.

CA ERwin DM includes a Windows-based graphical user interface, entity-relationship (ER) diagramming tools, custom editors, and a Model Explorer. The tools and editors help you define physical database objects and the Model Explorer provides a text-based view of model objects.

You can also use CA ERwin DM to streamline the application development process. Data Architects, Business Analysts, Data Modelers, and so on, can work independently on different parts of a model or different model types. The groups can collaborate and synchronize their work as well.

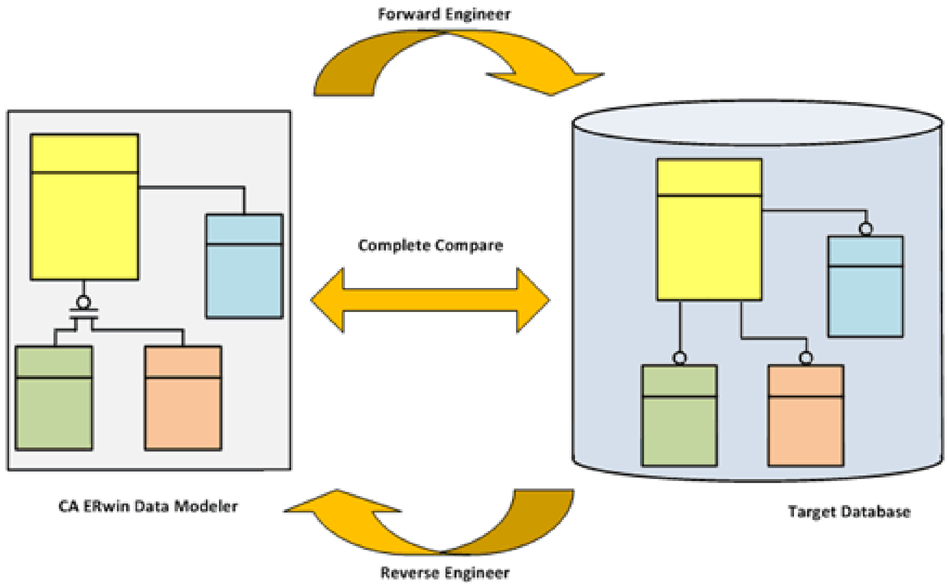
About CA ERwin Data Modeler

CA ERwin DM lets you create and maintain your relational database and the logical and physical models that describe it. CA ERwin DM works by providing a design solution that helps you create a visual blueprint (a data model) for your organization.

CA ERwin DM helps you design a logical data model to capture business rules and requirements. The application also supports designing or deriving a corresponding physical data model for your target server. You can then forward engineer the physical data model and automatically generate the physical database structures to your system catalog.

CA ERwin DM supports reverse engineering of existing databases and provides both a physical and a logical/physical data model. The forwarding engineering and reverse engineering options help you maintain an existing database, or migrate from your current database to a different one.

The Complete Compare feature automates the model and database synchronization by letting you compare the model with the database, displaying, and analyzing the differences. The comparison result enables you to move selected differences into the model or generate them into the database.



The Application Development Cycle

To create a data model, first the business rules are gathered and the logical constructs are created. Next, the physical design is created, and then a database is implemented. CA ERwin DM supports this iterative process by supporting multiple platforms, reuse of objects, and synchronization of changes between data models.

Chapter 2: Customizing the User Interface

This section contains the following topics:

Customize Toolbars

Configure your work environment to display or hide a toolbar or display or hide the options on a toolbar. You can create a toolbar and can add only the options that you use frequently. You can also modify an existing toolbar to display or hide a subset of icons. For example, if you rarely use the underline and strikeout text formatting options on the Formatting Toolbar, you can hide those options.

Similarly, you can dock a toolbar in a specific position or keep it floating and move it as you want.

Follow these steps:

1. To modify an existing toolbar, follow these steps:
   a. Click the arrow-down icon on the icon toolbar that you want to customize.
   b. Select Add or Remove Buttons, <icon toolbar name> to display the currently selected icons.

      Your changes are reflected immediately in the CA ERwin Data Modeler Workspace.
2. To create a toolbar, follow these steps:
   a. Click Tools, Customize.

      The Customize dialog opens.
   b. Go to the Toolbars tab and click New.
   c. Enter the name for your new toolbar.

      This text displays in the title bar for the new toolbar, when you select to undock it.
   d. Click OK.

      Your new toolbar name appears in the list of toolbars in the Customize dialog.

e. Select the toolbar name, then click the Commands tab.

f. Select a category, and drag the command icon out of the dialog to your new toolbar.

Your new toolbar is populated with the commands that you choose.

Customize Display Levels

The Display Toolbar includes the options to change the display levels in your model. The following display levels are available:

- Entity or table
- Primary key
- Keys
- Attribute or column
- Comments

Display level is changed to suit the type of model being created or the audience for whom the model is being presented.

You can change the display level in two ways:

- Select an object or a group of objects, then click the desired icon to apply the display setting to those objects.
- Do not select any objects, then click the icon to apply the display setting to the entire diagram. If you have already set display options for individual objects, you can override those settings for the entire diagram with this option.

Chapter 3: Setting Model-Level Default Values

This section contains the following topics:

Set Default Null Options, Data Types, Styles, and Object Owners

Use the Defaults tab in the Model Editor dialog to choose default values for null options, data types, themes, and object ownership.

Follow these steps:

1. Click Model, Model Properties.

   The Model Editor opens.

2. Click the Defaults tab and work with the following options:

   Non-Key Null Option

   Specifies whether you can have null as a value for a non-key attribute or column.

   Non-Identifying Relationship

   Specifies whether you can have null as a value for an attribute or a column that has a non-identifying relationship.

   Logical Datatype

   Specifies the data type that is used as the default for all attributes. You can change the data type for a specific attribute in the Attributes Editor dialog. You can also select whether the data type is inherited (⧉) or overridden (✎).

   Note: When you choose to inherit a data type, the value is inherited from the default settings. When you choose to override, the default value can be overridden and a different value can be added.

   Physical Datatype

   Specifies the default data type for all columns. You can change the data type for a specific column in the Columns Editor. You can also select whether the data type is inherent or overridden.

   Default Theme

   Specifies the default theme for the model. Your theme includes all the font, color, and formatting options for model objects. You use the Theme Editor to work with the formatting options.

   Use Data Type in Unification

   Specifies whether you want to use the data type when you unify three or more entities.

   Set Default Owners

   Opens the Set Default Owner dialog where you can set a global default for all objects, or for individual objects. You enter an owner name, and can optionally reset the ownership for all objects when you click Set Owner For All Objects.

Important! Although you can quickly reset the ownership for all objects using this feature, you cannot undo this operation.

3. Click Close.

   The default selections are set for the model and the Model Editor closes.

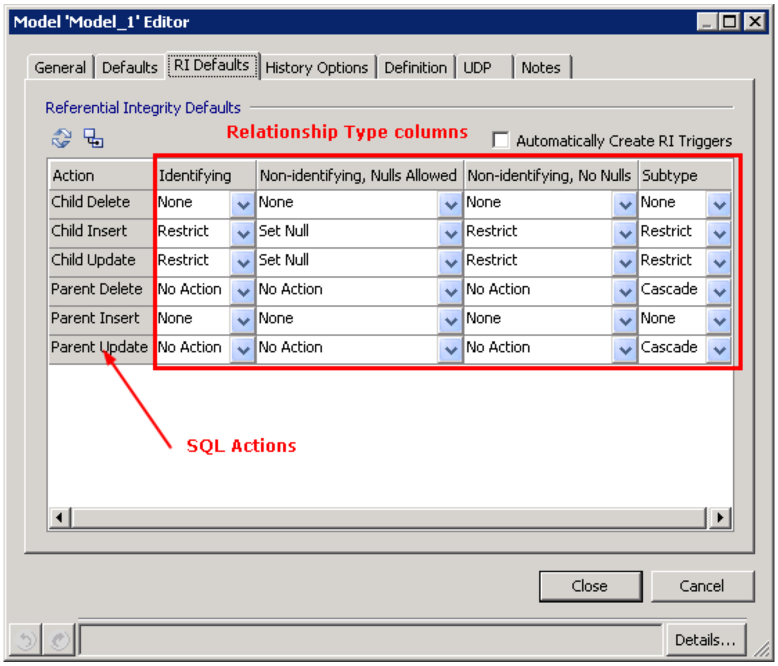
Set Default Referential Integrity Rules

Use the grid in the RI Defaults tab of the Model Editor to specify default referential integrity rules for relationships in your model.

Follow these steps:

1. Click Model, Model Properties.

   The Model Editor opens.



Relationship Type columns

□ Automatically Create RI Triggers

Action	Identifying	Non-identifying, Nulls Allowed	Non-identifying, No Nulls	Subtype
Child Delete	None	None	None	None
Child Insert	Restrict	Set Null	Restrict	Restrict
Child Update	Restrict	Set Null	Restrict	Restrict
Parent Delete	No Action	No Action	No Action	Cascade
Parent Insert	None	None	None	None
Parent Update	No Action	No Action	No Action	Cascade

SQL Actions

2. Click the RI Defaults tab and work with the following options:

   Rebind referential integrity settings on all relationships

   Applies the new default settings to any existing relationship in the model. If you change the default settings in an existing model, use this button to implement the new settings in the model for all existing relationships. If the new settings are saved, but not rebound, the defaults affect the new relationships only.

   Reset all Referential Integrity default settings to their original values

   Clears your settings and returns to the system defaults.

   Automatically Create RI Triggers

   Specifies whether you want to create triggers for the referential integrity defaults automatically.

   Action

   Lists a SQL action in each of its rows. For each SQL action, select the default RI action for each of the following relationship type columns:

Identifying

Lets you specify the RI setting for each of the SQL actions in the Action column. This setting applies to all identifying relationships in the model.

Non-identifying, Nulls Allowed

Lets you specify the RI setting for each of the SQL actions in the Action column. This setting applies to all non-identifying relationships in the model for which nulls are allowed.

Non-identifying, No Nulls

Lets you specify the RI setting for each SQL action in the Action column. The setting applies to all non-identifying relationships in the model for which nulls are not allowed.

Subtype

Lets you specify the RI setting for each of the SQL actions in the Action column. This setting applies to all subtype relationships in the model.

Note: Some target databases, such as Oracle, use a different terminology for RI trigger actions. Use the RI setting appropriate for your target database. For NONE and NO ACTION, use of NONE means no restrictions (no RI trigger is created). NO ACTION is identical to RESTRICT. In other words, the actions are allowed only if the conditions are met.

3. Click Close.

   Default referential integrity rules are set and the Model Editor closes.

Specify History Options for Model Objects and Events

Use the History Options tab in the Model Editor to specify history preferences for common model objects and events. Maintaining history information is helpful if you want to review the history of actions that occur in your model.

Follow these steps:

1. Click Model, Model Properties.

   The Model Editor opens.

2. Click the History Options tab and work with the following options:

   History Objects

   Displays a list of the common model objects. Select the check box for each object for which you want to save history.

   History Events

   Displays a list of the model events. Select the check box for each event for which you want to save history.

3. Click Close.

   The history options are specified for the model and the Model Editor closes.

Assign Model UDP Values

After you create user-defined properties (UDPs) for a model, you can specify them in the UDP tab of the Model Editor.

Follow these steps:

1. Click Model, Model Properties.

   The Model Editor opens.

2. Click the UDP tab and work with the following options:

   ☰

   Opens the User Defined Properties Editor so you can add a UDP or can edit an existing one.

   ⧉

   Resets the selected property to restore the inheritance.

   Note: This option is available only for property names that are inherited. In addition, this option becomes available only when you change the property value.

   Property

   Displays the name of the user-defined property. When you hover over a property name with your cursor, the UDP description displays as a tooltip.

   Value

   Specifies the current value that is assigned to the UDP. You can perform any of the following actions to change the value:

   - Enter the property value in the cell for *Integer*, *Real Number*, and *Text* data types.
   - Enter a date value for a *Date* data type.
   - Select a value from the Value list for the *List* data type. For example, you can add a UDP for a table to select a backup frequency such as *daily* from the Value list.
   - Enter a system command or click ... to browse for a file or executable program for the *Command* data type. For example, you can enter C:\Myfile.doc or can select it to launch the Microsoft Word document *Myfile.doc*.

3. Click Close.

   Model UDP values are assigned and the Model Editor closes.

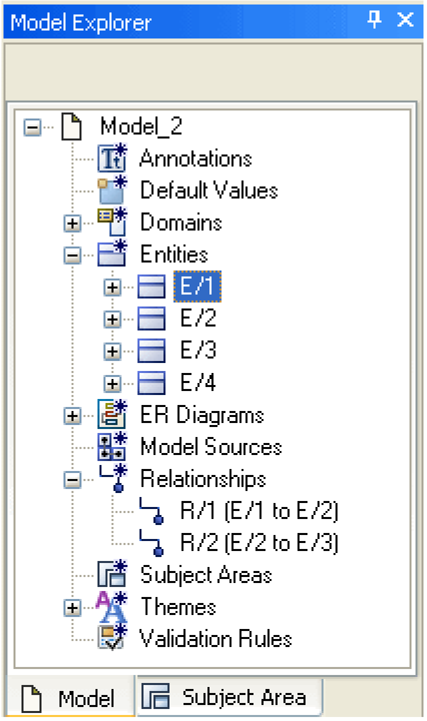
Chapter 4: Using the Model Explorer

This section contains the following topics:

Overview of Model Explorer

The Model Explorer provides an organized, hierarchical, text-based view of your data model and its contents. It provides an easy method for creating objects. The Model Explorer lets you create, display, navigate, and modify your model.

The Model Explorer has two panes, Model and Subject Area, as shown in the following illustration:



All the objects in your model are listed in the Model Explorer, but display differently based on the active pane.

- The Model pane includes every object in your model including subject areas and domains. The Model pane is the default view.
- The Subject Areas pane displays model objects that are sorted by subject area.

When you change an object in the Model Explorer, the graphical view of the model is immediately updated with the same change. For example, if you rename a table in the Model Explorer, the new table name appears in the Diagram Window and the relevant editors. Similarly, if you change an object the diagram or in an editor, you immediately see the change in the Model Explorer.

Just as you can create an entity, a table, or a relationship using the Toolbox, you can create an object in the Model Explorer. To create an object, right-click on any object and click New. You can drag some objects from the Model Explorer onto the Diagram Window to create an object. For example, you can drag a domain onto an attribute or a column.

Using the Model Explorer, you can open the property editor for an object in two ways:

- Right-click the object and click Properties.
- Select the object and click the Property Editor icon in the Model Explorer toolbar.

Note: For bulk editing of object properties, you can use the Bulk Editor.

The Model Explorer and the data model are always in sync, regardless of the method you choose to add or edit an object.

You can also move, copy, and delete most objects in the Model Explorer, same as you do in a Diagram window.

After you move, copy, or delete an object, all the instances in the model that reference the object are automatically updated.

Model View

The primary purpose of the Model view is model exploration and to function as an alternative launching point for object and property editors. The Model view lists all the object types that appear in the current data model. The object types that are displayed here are based on the model type and the target server.

For example, if the model type is Logical, the Model Explorer does not include physical objects such as a view. Similarly, if the model type is Physical, the Model Explorer does not include logical objects. If the target server is Oracle, then objects specific to Oracle such as Clusters are displayed.

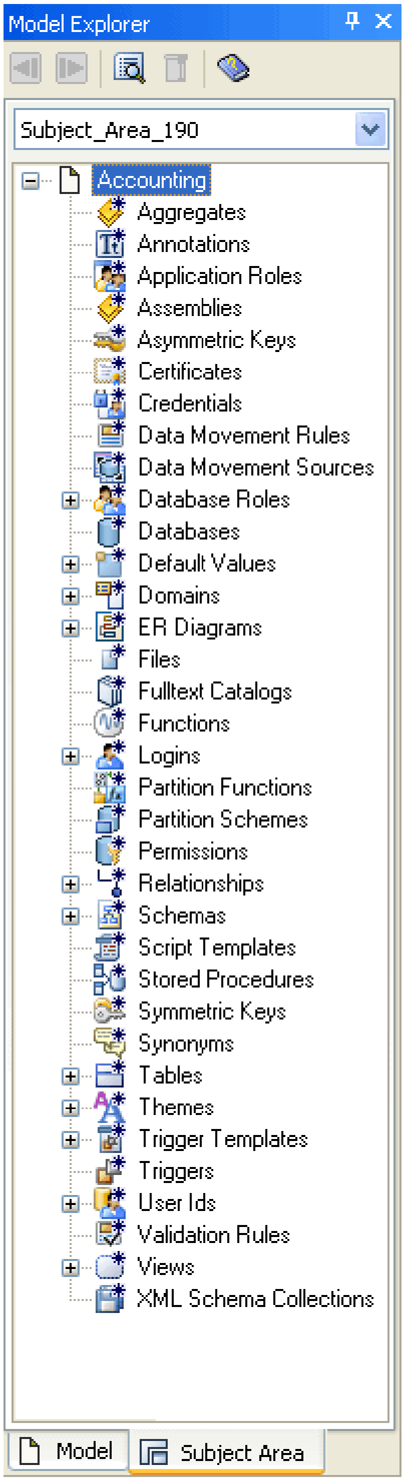
The following types of objects are not displayed in the Model Explorer:

- Minor objects that are included in the parent object editor. For example, a Valid Value object is created, deleted, and edited by the Validation Rule Editor.
- An object that represents configuration information, created for an internal process, and is not actual model data. For example, as a Complete Compare Option Set.

Subject Area View

The Subject Area view provides a filtered view of the model and minimizes unnecessary tree information. This view displays the model objects that are sorted by Subject Area. You can create Subject Areas to divide the model into smaller manageable parts, which is helpful when you work with large models.

The following illustration shows the Subject Area view:



Subject_Area_190

- Accounting
  - Aggregates
  - Annotations
  - Application Roles
  - Assemblies
  - Asymmetric Keys
  - Certificates
  - Credentials
  - Data Movement Rules
  - Data Movement Sources
  - Database Roles
  - Databases
  - Default Values
  - Domains
  - ER Diagrams
  - Files
  - Fulltext Catalogs
  - Functions
  - Logins
  - Partition Functions
  - Partition Schemes
  - Permissions
  - Relationships
  - Schemas
  - Script Templates
  - Stored Procedures
  - Symmetric Keys
  - Synonyms
  - Tables
  - Themes
  - Trigger Templates
  - Triggers
  - User Ids
  - Validation Rules
  - Views
  - XML Schema Collections

Select a Subject Area to view the objects that are included in it. For example, in the previous illustration you see that the tables are categorized as tables, and views are categorized as views. This view is different from the display of the Subject Area objects in the Model view. Only those relationships are included here that are visible on the diagram and have both the endpoints present in the Subject Area.

Note: The global objects that do not relate to the selected Subject Area, such as Subject Area objects or model source objects are not displayed.

Create a Subject Area

You can create a subject area using the Subject Area Editor.

Follow these steps:

1. Click Model, Subject Areas.

   The Subject Area Editor opens.

2. Click New.

   A new subject area with the default name is added to the Navigation Grid.

3. Work with the following options:

   Name

   Displays the subject area name.

   Auto-populate

   Specifies to populate the selected subject area with all the model objects automatically.

4. Click the General tab and specify common properties for the selected subject area.
5. Click the Members tab and specify the members for the selected subject area.
6. (Optional) Click the Definition tab to enter the definition text.
7. (Optional) Click the UDP tab to add subject area user-defined properties.
8. Click Close.

   A new subject area is created and the Subject Area Editor closes.

Rename Objects

Every object in your model is represented in the Model Explorer by its name. Just as you can rename an object using on-diagram editing in the Diagram Window, you can rename an object directly in the Model Explorer.

Follow these steps:

1. Click the object name, and then press F2.

   An edit box opens around the object name.

2. Enter the new object name.

   The new name of that object appears throughout the Model Explorer and the model.

Note: Any unique name rules that you select for your model are enforced when you rename an object in the Model Explorer.

Select Unique Name Rules

Unique name rules define how you want to respond to a duplicate entity or table name in the data model.

Follow these steps:

1. Click Actions, Model Naming Options.

   The Model Naming Options dialog opens.

2. Click the Duplicate Names tab and set preferences for duplicate names.
3. Click OK.

   Your changes are saved and the Model Naming Options dialog closes.

The Go to Diagram Option

When you have several hundreds of objects in a model, it might be difficult to locate a specific object. To locate an object in a diagram quickly, right-click the object in the Model Explorer and click Go To Diagram. The object is selected in the Diagram window.

Add a Domain in the Model Explorer

The Model view lists the domains for the current model, which include the default domains and the domains created by you.

Follow these steps:

1. Right-click a domain type (blob, string, number, and so on) and click New.

   An edit box opens with a default domain name.

   Note: The selected domain type is assigned as the parent.

2. Enter a name for the domain in the text box and press Enter.

   The new domain appears according to the sort order that you specify (either alphabetically or hierarchically). To switch the sort order of the Domains, right-click the Domains folder, and select the sort option that you prefer.

Note: For more information about domains, see Working with Data Models.

Add a Model Source in the Model Explorer

When you work with design layers, you link a model with its model source to identify the historical parent of the model. After you link a data model to its source, you can synchronize the two models to apply changes to one or the other selectively.

Follow these steps:

1. Right-click Model Sources and click New.

   The Add Model Source wizard opens.

2. Follow the steps in the Add Model Source wizard to finish adding the model source.

3. Click Add to complete the process and close the wizard.

   The source objects are added to your target model, and models are linked. As you continue to work with the models, all the changes you apply to the model and the model source are tracked.

Note: For more information about adding model sources, see Working with Design Layers.

Chapter 5: The Theme Editor

A theme is a collection of font and color settings and is applied to a model, an object, a group of objects, or a subject area. Themes help you achieve the following objectives:

- Maintain the organizational design standard
- Denote the development lifecycle of a model
- Customize the designs in a model for various end users, for example, Human Resource or Marketing departments

Themes are applied in the following hierarchical order:

- Model - to replace the default theme that a new model uses.
- Subject Area - to override the theme of the model.
- Diagram - to override the theme of its owner (Model or Subject Area).
- Entity, View, Cached View, Annotation, or Attribute - to override the theme of the diagram.
- Drawing Object - to override all other themes. However, you cannot do an override on an attribute because attributes have no drawing objects. At the drawing object level, individual characteristics of a theme can be overridden once you select a theme.

You can open the Theme Editor in any of these ways:

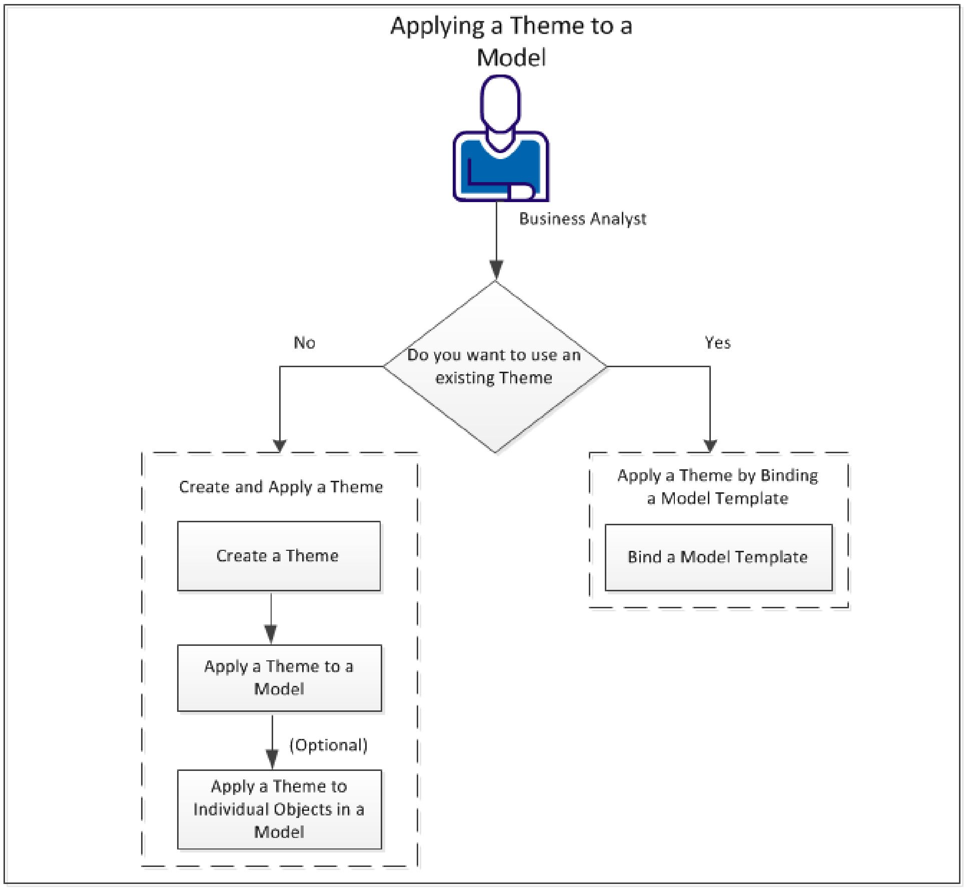
- From the General tab on the Diagram Editor
- When you click Themes on the View menu
- From the Model Explorer

The application ships with two themes, Default and Classic. You cannot edit these themes, but you can add new themes. You can also override the settings that are defined in a theme, in the Style tab of the individual editors.

For example, you are a Business Analyst in a software company. You want to do the following tasks:

- Maintain an organizational design standard in the Employee database.
- Organize the Employee database according to the departments in the company.

You identify all the departments in the company and create a theme for each department. You apply the themes to all of the departments to identify each department and its employees. This example is used consistently to explain all the procedures that are involved in creating and applying a theme.


Applying a Theme to a Model
Business Analyst
Do you want to use an existing Theme
No
Yes
Create and Apply a Theme
Create a Theme
Apply a Theme to a Model
(Optional)
Apply a Theme to Individual Objects in a Model
Apply a Theme by Binding a Model Template
Bind a Model Template

Perform the following tasks to apply a theme:

- Do you want to use an existing theme (see page 30)
- Create and apply a theme (see page 30)
  1. Create a theme (see page 31)
  2. Apply a theme to a model (see page 41)
  3. (Optional) Apply a theme to individual objects in a model (see page 42)
- Apply a theme by binding a model template (see page 42)
  1. Bind a Model Template (see page 43)

Do you want to use an existing theme

If you have a standard theme in your organization, you can use that theme in all your models by binding it to a model template. Using a standard theme ensures consistency across all your models.

If you do not have any predefined themes, you can create a theme and can apply it to your models.

Create and Apply a Theme

If you do not have a standard theme in your organization, you can create one and can apply it for all your models. You can also create multiple themes and apply them to different objects in your model.

Create a Theme

After you open the Employee data model, identify the company departments and create a theme for each object or model.

When you create a theme, you can select and apply the theme to any object or model in the Theme Editor. For example, suppose that your Employee data model has separate objects for part-time and full-time employees working in the same department. You can apply a theme to the part-time employee object to differentiate it from the remaining objects in the Employee data model.

You use the Defaults tab to set the preferences for font and fill colors for all objects in your model. When you select a theme for your model, all objects in your model inherit these values.

Follow these steps:

1. Open CA ERwin Data Modeler and then open a data model.
2. Expand Model Explorer, right-click Themes, and click New.

   A new theme is created with a default name.
3. Right-click the new theme and click Properties.

   The Theme Editor opens.
4. (Optional) Edit the name of the theme.
5. Edit the properties in each column for the new theme.
6. Click the Defaults tab and make your selections for the desired options as appropriate for your organization.
7. (Optional) Customize your theme by selecting the desired options on the other tabs as appropriate for your organization.
8. Click Close.

   A theme is created and the Theme Editor closes.

Set Display Preferences for Entities and Tables

Use the Entity or the Table tab in the Theme Editor to set the preferences for displaying name and other definitions of entities or tables. The Entity tab is displayed when you view a Logical model and the Table tab is displayed when you view a Physical model.

Follow these steps:

1. Click View, Themes.

   The Theme Editor opens.

2. Select the theme for which you want to set the display preferences, and click the Entity or the Table tab.

3. Work with the following options:

   Name Display

   Font Options

   Specifies the default font styles to display the name of an entity. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Fill Options

   Specifies the default fill option for an entity. Click 🪣 to change the fill style details. You can also choose to inherit, override, or harden this setting.

   Outline Options

   Specifies the default outline styles for an entity. Click ⬚ to change the outline style details. You can also choose to inherit, override, or harden this setting.

   Definition Display

   Font Options

   Specifies the default font styles to display the entity definition. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Fill Options

   Specifies the default fill option for the entity definition. Click 🪣 to change the fill style details. You can also choose to inherit, override, or harden this setting.

   Outline Options

   Specifies the default outline styles for the entity definition. Click ⬚ to change the outline style details. You can also choose to inherit, override, or harden this setting.

   Your selections are displayed in the Sample section.

4. Click Close.

   Your changes are saved and the Theme Editor closes.

Note: You cannot select to inherit, override, or harden any display setting for the Default theme.

Set Display Preferences for Owned Attributes or Columns and Foreign Keys

Use the Attribute or the Column tab in the Theme Editor to set the preferences for displaying attributes, columns, and foreign keys. The Attribute tab is displayed when you view a Logical model and the Column tab is displayed when you view a Physical model.

Follow these steps:

1. Click View, Themes.

   The Theme Editor opens.

2. Select the theme for which you want to set the display preferences, and click the Attribute tab.

3. Work with the following options:

   Owned Attribute Font

   Font Options

   Specifies the default font for entity attributes. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Foreign Key Inheritance

   Font Inheritance Type

   Specifies the type of inheritance for the font of a foreign key. You can choose the inheritance to be based on the primary key, the relationship, or no inheritance.

   Color Inheritance Type

   Specifies the type of inheritance for the color of a foreign key. You can choose the inheritance to be based on the primary key, the relationship, or no inheritance.

   Foreign Key Attribute Font

   Specifies the font of the foreign key attribute. Click A to select the font. You can also choose to inherit, override, or harden this setting.

   Foreign Key Attribute Color

   Specifies the color of the foreign key column. Click 🪣 to select the color. You can also choose to inherit, override, or harden this setting.

   Your selections are displayed in the respective Sample sections.

4. Click Close.

   Your changes are saved and the Theme Editor closes.

Note: You cannot select to inherit, override, or harden any display setting for the Default theme.

Set Display Preferences for Keys

Use the Key tab in the Theme Editor to set the preferences for displaying the alternate keys and inversion entities.

Follow these steps:

1. Click View, Themes.

   The Theme Editor opens.

2. Select the theme for which you want to set the display preferences, and click the Key tab.

3. Work with the following options:

   Key Display Preference

   Style Preference

   Specifies whether to show the alternate key (AK) style or the inversion entity (IE) style preference when both apply.

   Alternate Key (AK) Column Font

   Font Options

   Specifies the default font for the alternate key column. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Inversion Entity (IE) Column Font

   Font Options

   Specifies the default font for the inversion entity column. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Your selections are displayed in the respective Sample sections.

4. Click Close.

   Your changes are saved and the Theme Editor closes.

Note: You cannot select to inherit, override, or harden any display setting for the Default theme.

Set Display Preferences for Views and Materialized Views

Use the View or the Materialized View tab in the Theme Editor to set the preferences for displaying the views and materialized views. The Materialized View tab is displayed only for the databases that support materialized views.

Follow these steps:

1. Click View, Themes.

   The Theme Editor opens.

2. Select the theme for which you want to set the display preferences, and click the View tab.

3. Work with the following options:

   Name Display

   Font Options

   Specifies the default font styles to display the name of a view. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Fill Options

   Specifies the default fill option for a view. Click 🪣 to change the fill style details. You can also choose to inherit, override, or harden this setting.

   Outline Options

   Specifies the default outline styles for a view name. Click ⬚ to change the outline style details. You can also choose to inherit, override, or harden this setting.

   Definition Display

   Font Options

   Specifies the default font styles to display the view definition. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

   Fill Options

   Specifies the default fill color for the view definition. Click 🪣 to change the fill style details. You can also choose to inherit, override, or harden this setting.

   Outline Options

   Specifies the default outline styles for the view definition. Click ⬚ to change the outline style details. You can also choose to inherit, override, or harden this setting.

   Your selections are displayed in the Sample section.

Column/Attribute Display Font

Owned Column/Attribute

Specifies the default font style in an owned column or attribute. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

Expression

Specifies the default font style in an expression. Click A to change the font style details. You can also choose to inherit, override, or harden this setting.

The following fields are displayed only in the Materialized View tab:

Alternate Key (AK) Column/Attribute (for MV only)

Specifies the default font style in an alternate key column or attribute. Click to change the font style details. You can choose to make the font inherit, override, or harden.

Inversion Entity (IE) (for MV only)

Specifies the default font style in an inversion entry column or attribute. Click to change the font style details. You can choose to make the font inherit, override, or harden.

4. Click Close.

   Your changes are saved and the Theme Editor closes.

Note: You cannot select to inherit, override, or harden any display setting for the Default theme.

Set Display Preferences for Subtype Symbol

Use the Subtype Symbol tab in the Theme Editor to set the preferences for displaying the subtype symbols.

Follow these steps:

1. Click View, Themes.

   The Theme Editor opens.

2. Select the theme for which you want to set the display preferences, and click the Subtype Symbol tab.

3. Work with the following options:

   Style Inheritance

   Inherit Discriminator font from Supertype name font

   Specifies that the font is inherited from the Supertype name font. If you select this check box, you cannot edit the Font Options field in the Subtype Symbol Text Font section.

   Inherit text color from line color

   Specifies that the text color is inherited from the line color. If you select this check box, you cannot edit the Color field in the Subtype Symbol Text Color section.

   Inherit Discriminator color from Supertype outline color

   Specifies that the line color is inherited from the parent Supertype outline color. If you select this check box, you cannot edit the Color field in the Subtype Symbol Style section.

   Subtype Symbol Text Font

   Font Options

   Specifies the font that you use for the subtype symbols in your model. If you have selected the Inherit Discriminator font from Supertype name font check box, you cannot edit this field.

   Subtype Symbol Text Color

   Color

   Specifies the text color that you use for the subtype symbols in your model. If you have selected the Inherit text color from line color check box, you cannot edit this field.

   Subtype Symbol Style

   Line Width

   Specifies the default line width for subtype symbols. Use the spin controls to increase or decrease the line width.

Color

Specifies the default color for subtype symbol lines. If you have selected the Inherit Discriminator color from Supertype outline color check box, you cannot edit this field.

Fill Options

Specifies the default line fill options for subtype symbols. Click to change the fill style details.

Note: You can select to inherit, override, or harden any of these settings except for display settings for the Default Theme.

4. Click Close.

   Your changes are saved and the Theme Editor closes.

Set Display Preferences for Annotations and Shapes

Use the Annotation/Shape tab in the Theme Editor to set the font and color of the annotation text. Use this tab to change the display properties of annotations and shapes.

Follow these steps:

1. Click View, Themes.

   The Theme Editor opens.

2. Select the theme for which you want to set the display preferences, and click the Annotation/Shape tab.

   Font Options

   > Specifies the default font styles for the selected drawing object. Click to change the font style details. You can also choose to inherit, override, or harden this setting.

   Vertical Alignment

   > Specifies the vertical alignment position of the text in the annotation. You can also choose to inherit, override, or harden this setting.

   Horizontal Alignment

   > Specifies the horizontal alignment position of the text in the annotation. You can also choose to inherit, override, or harden this setting.

   Text Handling Style

   > Specifies how you want the annotation to adjust itself when you enter text. You can also choose to inherit, override, or harden this setting.

   Fill Options

   > Specifies the default fill color for the selected annotation. Click to change the fill style details. You can also choose to inherit, override, or harden this setting.

   Line Options

   > Specifies the default line styles for the selected annotation. Click to change the outline style details. You can also choose to inherit, override, or harden this setting.

   Outline Options

   > Specifies the default outline styles for the selected annotation. Click to change the outline style details. You can also choose to inherit, override, or harden this setting.

   Click OK.

   The new style for the annotation is saved.

Apply a Theme to a Model

Apply the theme to the Employee data model and maintain the organizational design standard. When you apply a theme to a model, all new objects inherit the customized values that you have defined in the theme.

Follow these steps:

1. Click Model, Model Properties.

   The Model Editor opens.

2. Click the Defaults tab.

3. Click the Default Theme drop-down list and select a theme.

   Your theme is applied to the model.

4. Click Close.

   The Model Editor closes.

You have successfully applied a theme to your model.

If you want to apply a different theme to the individual objects in your model, see the next topic. You do not have to close the model to apply a different theme.

(Optional) Apply a Theme to Individual Objects in a Model

You can apply a theme to individual objects instead of the entire model. In your Employee data model, you can apply a different theme to each department (object) to organize the model. For example, you can apply one theme to the Human Resources object and another theme to the Marketing Division object.

You can also apply a theme by copying it from one object and pasting it to another object using the context menu. You can copy and paste a theme between similar objects and not between different objects. For example, you can copy and paste a theme between entities and not between an entity and a diagram. This option is useful if you want to apply a theme including any overrides that you have applied for a specific entity.

Follow these steps:

1. Ctrl + click each object that you want to select in your model.
2. Right-click and click Override Fonts and Colors.

   The Multiple Shape Object Editor opens.
3. Click the Style tab.
4. Click the Theme drop-down list under the Theme for Default Font and Color options section, and select your theme.

   Your theme is applied to the selected objects.
5. Click Close.

   The Multiple Shape Object Editor closes.

The theme is applied to the selected objects.

Apply a Theme by Binding a Model Template

If you have a standard theme in your company, you can bind a model template having the standard theme to any new data model. When you bind the model template, the current model inherits the theme from the model template.

Bind a Model Template

Binding a model template to a data model copies the non-built-in objects from the template model to the target model. When you bind a template to a model, the predefined objects populate your model. This process eliminates the need to create objects manually or to use reverse engineering. Changes to the model template are displayed in the models that use it, maintaining consistency of your object names. When you open a model that is bound to a model template, you can choose to synchronize any model template changes with current the model. Whenever you open a model that is bound to a model template, confirm whether to synchronize any model template changes with the objects used in the model.

You can bind model templates any time, while creating a model or after the model is created.

Objects that are brought in to a model from a template fully follow the modeling rules. If any conflicts occur in the model, the existing data is preserved as much as possible.

Follow these steps:

1. Open CA ERwin Data Modeler and then open a data model.
2. Click File, Model Templates, Bind Template.

   The Bind Model Template dialog opens.
3. Use the toolbar to browse for the template file locally on your computer or in the Mart.
4. Select one of the following options and then click OK:

   Load the entire contents of the template

   > Specifies to load the entire contents of the template. This option binds all of the model template objects to the current model without the opportunity to review and select individual template objects.

   Run the Template Wizard to select objects for synchronization

   > Specifies to open the Model Template Synchronization Wizard so that you can select the objects in the model template to synchronize with the data model.

   The model opens in the workspace with the template objects copied into it. The theme is applied to the data model.

Chapter 6: Creating and Using Domains

This section contains the following topics:

Overview of Domains

A *domain* is a model object that you can use to assign properties quickly to an attribute or column. By using domains, you promote consistency because you can reuse a domain as many times as you like in a single or multiple data models.

Domains also reduce the time that is spent on development and maintenance. If you change the domain, all attributes or columns that are associated with the domain also change.

Domains have two types of properties:

- *Non-inheritable properties:* The non-inheritable properties do not migrate to child domains or attributes and columns that are associated with the domain. This is because they are properties of the domain itself.
- *Inheritable properties:* The inheritable properties migrate to child domains and to the attributes and columns that are associated with the domain.

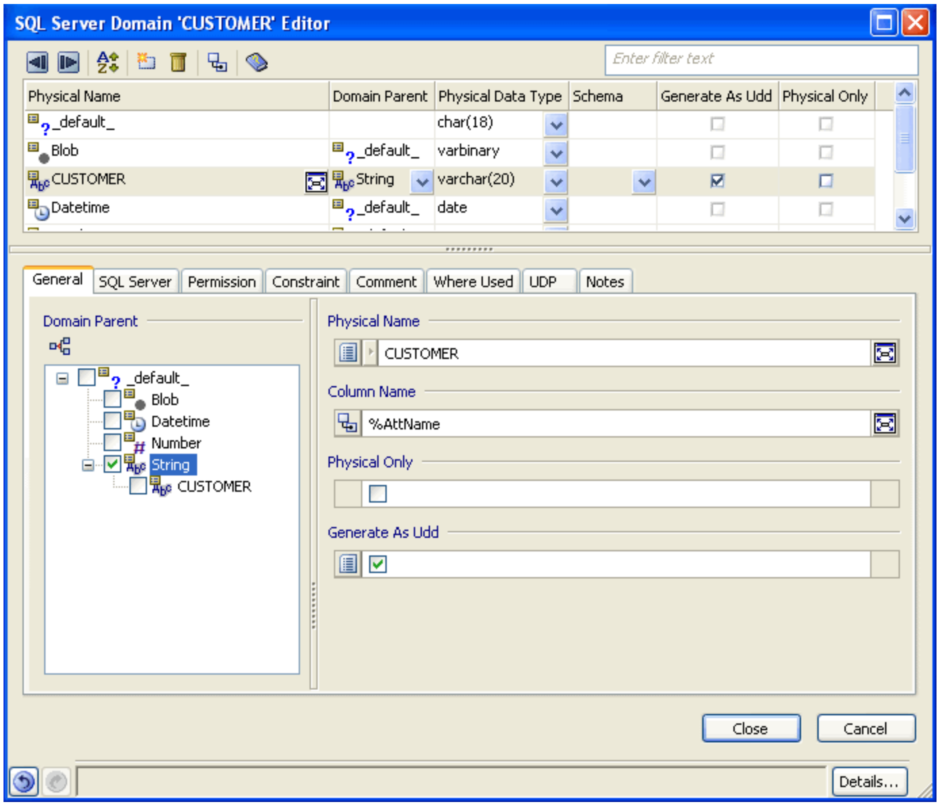
The inheritable and non-inheritable domain name properties appear on the General tab. They also appear on the target database tab (for example, SQL Server) in a physical model.

When you drag-and-drop a domain from the Model Explorer onto an entity or a table, you add an attribute or a column. The new attribute or column inherits all the inheritable properties, including name, from the domain.

In the Model Explorer and in the Domain Editor, a unique icon represents each domain. A default icon is used for new domains, but you can choose a different icon to represent the domains you create.

Domain Editor

The Domain Editor lists all the domains for the data model. You can arrange them hierarchically or alphabetically. The Model Explorer and the Domain Editor list the same domains and lets you sort the domains in hierarchical or alphabetic order. You can create and modify both physical and logical domains using the Domain Editor.



SQL Server Domain 'CUSTOMER' Editor

Enter filter text

Physical Name	Domain Parent	Physical Data Type	Schema	Generate As Udd	Physical Only
default		char(18)		□	□
Blob	_default_	varbinary		□	□
CUSTOMER	String	varchar(20)		☑	□
Datetime	_default_	date		□	□

Domain Parent

- □ _default_
  - □ Blob
  - □ Datetime
  - □ Number
  - ☑ String
    - □ CUSTOMER

Physical Name: CUSTOMER

Column Name: %AttName

Physical Only: □

Generate As Udd: ☑

The tabs and options in the editor change depending on whether the model is logical, physical, or logical/physical.

Domains include the following properties:

- Domain name and column name
- Column data type, permission, default value, and valid value
- Domain comment or note and column comment or name
- User-defined property
- User notes

Note: You cannot change the properties of the standard domains (*<default>*, *Blob*, *Datetime*, *Number*, and *String*).

Drag a Domain from the Model Explorer to Your Model

Use drag-and-drop to add a domain from the Model Explorer to an entity or a table in your model.

Follow these steps:

1. Expand the list of domains in the Model Explorer and click the domain that you want to add to your model.
2. Drag and drop the domain onto the entity or table in your model.

   The domain is added as an attribute or column.

   Note: Depending on the Unique Naming rules that apply to the data model, you are prompted to rename each attribute or column. You are prompted because, the same domain name is assigned to each new attribute or column automatically.

Creating Domains

Use the Domain Editor to create domains in addition to the default domains. You can create a domain while you are in a logical model (*logical edit mode*) or a physical model (*physical edit mode*). The options in the editor change depending on the edit mode.

Add a Domain in the Logical Edit Mode

The Domain Editor lets you define properties for attributes and columns. You can create a domain and can attach it to any number of attributes and columns instead of defining properties for each column individually. You can work with features in the Domain Editor to add a domain to the logical model.

Follow these steps:

1. Click Model, Domains.

   The Domain Editor opens.
2. Click the New button on the toolbar.

   A new domain with a default name is added and is placed in alphabetical order.

3. Work with the following options in the Navigation Grid:

   Name

   > Displays the name of the domain in the logical model.
   >
   > **Note:** Duplicate domain names are not allowed.

   Domain Parent

   > Displays the current parent domain. You can change the domain using the drop-down list.

   Logical Data Type

   > Specifies the logical data type for the domain. You can change the data type using the drop-down list. The drop-down list contains the data types related to the current domain.

   Logical Only

   > Specifies whether the domain appears only in the logical model. Select this check box if you want the domain to appear only in the logical model. Clear this check box if you want the domain in the physical model. If you select this check box in a logical-only model and you derive a new physical-only model, the domain does not appear in the new model.

4. Work with the other tabs in the editor to define more domain properties.
5. Click Close.

   The domain is added to the logical model and the Domain Editor closes.

Add a Domain in the Physical Edit Mode

You can work with the features in the Domain Editor to add a domain to the physical model.

Follow these steps:

1. Click Model, Domains.

   The Domain Editor opens.

2. Click the New button on the toolbar.

   A new domain with a default name is added and is placed in alphabetical order.

3. Work with the following options in the Navigation Grid:

   Physical Name

   > Displays the name of the domain in the physical model.

   Domain Parent

   > Displays the current parent domain. You can change the domain using the drop-down list.

Physical Data Type

Specifies the physical data type for the domain. You can change the data type using the drop-down list. The drop-down list contains the data types related to the current domain.

Anchored to Type

Specifies whether the domain is anchored to a column or a variable. This column is enabled only for a DB2 LUW user-defined data type.

Schema

Specifies the schema that is associated with the domain.

Distinct Type

Specifies whether the domain is of distinct type.

Physical Only

Specifies whether the domain appears only in the physical model. Select this check box if you want the selected domain to appear in the physical model only. Clear this check box if you want the selected domain in the logical model.

4. Work with the other tabs in the editor to define more domain properties.
5. Click Close.

   The domain is added to the physical model and the Domain Editor closes.

Assign a Data Type to a Domain

A *data type* is a domain property and defines the physical properties of a column in a database. You can define length (number of characters), type (alpha or numeric), and precision (decimal) properties. You can also define a logical data type, which usually corresponds to a physical data type.

By default, a new domain is assigned the same data type as its parent domain. You can change it at any time.

Follow these steps:

1. Click Model, Domains.

   The Domain Editor opens.

2. Select the domain in the Navigation Grid for which you want to assign a data type.
3. Select a new data type for the domain using the drop-down list in the Logical Data Type column (*logical edit mode*) or the Physical Data Type column (*physical edit mode*).
4. Click Close.

   The data type is assigned to the domain and the Domain Editor closes.

Assign or Change a Domain Icon

You can assign any image (.bmp file) as the icon for the non-inheritable property of the domain. You can assign the same image or can select a different image that the attributes or columns of the domain can inherit.

Note: You can change the domain icon only in the logical model.

Follow these steps:

1. Click Model, Domains.

   The Domain Editor opens.

2. Select the domain in the Navigation Grid that you want to work with and click the General tab.

3. Specify the image to use in the Image drop-down or click New to open the Image Editor and create one.

4. Click Close.

   The icon is assigned to the selected domain.

Chapter 7: Applying Naming and Datatype Standards

This section contains the following topics:

Applying Naming Standards

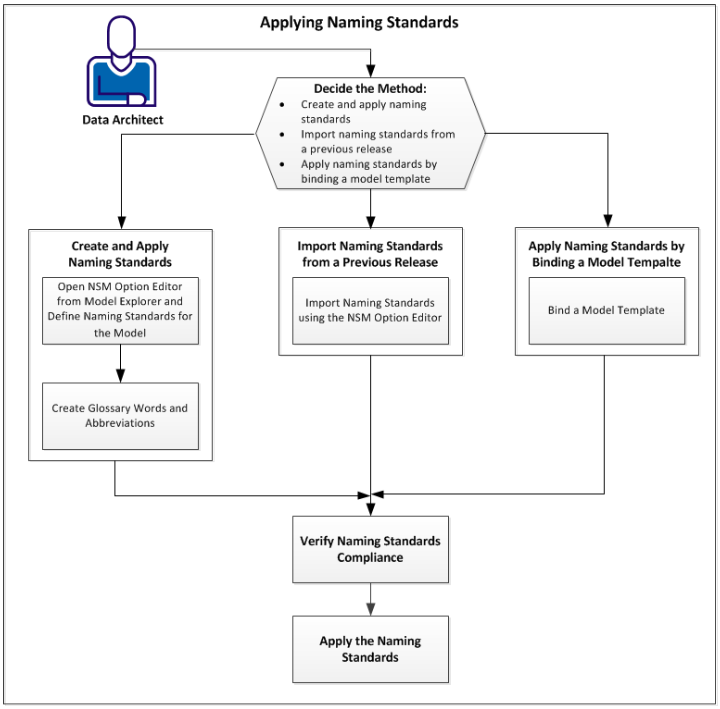
You create naming standards so that everyone in your organization uses names correctly and consistently. Use NSM Options Editor to create naming standards and develop a glossary that contains glossary words, their abbreviations, and alternate abbreviations.

You can create multiple NSM Option objects and can apply one of them to a model. Each NSM Option object includes the naming standards for logical and physical model objects, and the glossary. The components of logical and physical model names are prime, modifier, or class. To define naming standards for entities, tables, attributes, columns, and domains specify the components and the order in which to present them.

If you do not use naming standards, you may have modeling design problems. These design problems could produce errors when you run queries to extract data.

Note: Suppose that you upgrade a model from release r7 through r7.3.12 or from r8 through r8.2.9 to the latest release. If the source model includes NSM files, the naming standards are not included in the upgraded model. To get the naming standards, save the NSM file to your local computer, import it to a template model, and then bind the template to the upgraded model. We recommend that you do not include any other objects in the template model, other than the NSM object.

You can apply naming standards in three ways, as illustrated in the following diagram:


Applying Naming Standards
Data Architect
Decide the Method:
• Create and apply naming standards
• Import naming standards from a previous release
• Apply naming standards by binding a model template
Create and Apply Naming Standards
Open NSM Option Editor from Model Explorer and Define Naming Standards for the Model
Create Glossary Words and Abbreviations
Import Naming Standards from a Previous Release
Import Naming Standards using the NSM Option Editor
Apply Naming Standards by Binding a Model Tempalte
Bind a Model Template
Verify Naming Standards Compliance
Apply the Naming Standards

Follow these steps apply naming standards:

1. Decide the method. (see page 53)
2. Verify naming standards compliance. (see page 60)
3. Apply the naming standards. (see page 61)

Decide the Method

You can apply naming standards in the following ways:

- Create and apply: Use this option when you are creating the naming standards for the first time.
- Import from a previous version: Use this option if you have created naming standards using a previous version of CA ERwin DM.
- Bind a model template: Use this option if you want to use the naming standards that you have already created and attached to a model or a model template.

Create and Apply Naming Standards

You create naming standards if you are implementing standards for the first time.

Open NSM Option Editor from Model Explorer and Define Naming Standards for a Logical Model

You can specify naming standards for logical names that are different from the naming standards for physical names. Use the Logical tab in NSM Option Editor to apply naming standards to entities, attributes, and domains. In addition, use this tab to specify the order in which parts of names appear in the logical name.

Follow these steps:

1. Expand the Model Explorer, right-click NSM Options and click New.

   A new NSM Option object is added.

2. Right-click the new object and click Properties.

   NSM Option Editor appears.

3. Go to the Logical tab.
4. Select the part of the name such as prime, modifier1, modifier2, or class that you want to include first in the logical name. Specify the parts in the Entity, Attribute, and Domain rows.
5. Click Close.

   The data is saved.

Open NSM Option Editor from Model Explorer and Define Naming Standards for a Physical Model

You can specify naming standards for physical names that are different from the naming standards for logical names. Use the Physical tab in NSM Option Editor to apply naming standards to tables, columns, and domains. In addition use the editor and specify the order in which parts of names appear in the physical name.

Follow these steps:

1. Expand NSM Options in the Model Explorer and select the NSM Option object for which you want to add physical names.
2. Right-click and select Properties.

   NSM Option Editor appears.
3. Go to the Physical tab.
4. Select the part of the name such as prime, modifier1, modifier2, or class that you want to include first in the physical name. Specify the parts in the the Table, Column, and Domain rows.
5. Click Close.

   The data is saved.

Handle Vowels and Special Characters

If you do not want special characters and vowels in the name of your physical models, you can set actions to remove them.

Follow these steps:

1. Click Actions, Model Naming Options.

   The Model Naming Options dialog opens.

2. Click the Name Mapping tab.

3. Select *one* of the following actions from the Special Characters drop-down list:

   Leave

   Retains the special characters in the physical model object names.

   Remove

   Removes the special characters from the physical model object names.

   Replace

   Replaces the special character in the physical model object names with a specified character. Type the character in the Replace With box.

4. Select the Remove Vowels check box.

5. Click OK.

   The special characters and vowels are set for the Naming Standards in the physical model. The action that is taken for special characters and vowels is set when you update the physical naming standards.

Create Glossary Words and Abbreviations

You can enter words and their corresponding abbreviations directly into the glossary. You can enter, display, edit, and specify the word type (prime, modifier1, modifier2, or class).

Follow these steps:

1. Expand the Model Explorer, right-click NSM Options and click New.

   A new NSM Option object is added.

2. Right-click the new object and click Properties.

   NSM Option Editor appears.

3. Click the Glossary tab and work with the following options:

   Match whole words only

   Specifies whether to substitute glossary abbreviation on whole words only or to include parts of words or substrings. Select the check box to substitute glossary abbreviation on whole words only. Clear the check box to substitute glossary abbreviation for complete words, as well as on parts of words or substrings.

   Word Type

   Displays only the selected type of glossary words.

   Word/Words

   Specifies the glossary word or words that you want to abbreviate.

   Abbreviation

   Specifies the standard abbreviation.

   Alt Abb

   Specifies an alternate abbreviation.

   P, M1, M2, C

   Specifies whether you want to use the word as a prime, modifier 1, modifier 2, or class.

4. Click Close.

   The data is saved.

Import Words and Abbreviations into the Naming Standards Glossary

You can import words and abbreviations into the glossary from an existing naming standards CSV file. Be sure to examine the data in each row and edit or add as required.

Follow these steps::

1. Expand the Model Explorer, right-click NSM Options and click New.

   A new NSM Option object is added.

2. Right-click the new object and click Properties.

   NSM Option Editor appears.

3. Click the Glossary tab, then click Import.

   The Open dialog appears.

4. Select the path and file name of the CSV file, and click Open.

   The words and corresponding abbreviations are imported into the Naming Standards glossary.

Example

You have two tables, Customer and Accounts. The Customer table has First Name and Last Name columns.

Suppose you want to implement the following standards:

- Prefix the names of the tables with Tbl_.
- Display First Name as Col_FNAME and Last Name as Col_LNAME.

How you do it:

1. Add an NSM Object and make it active.
2. In the Logical tab, for Entity and Attribute, select Modifier 1 in the Part 1 and Prime in the Part 2 column.
3. In the Physical tab, for the Table and Column object types, select Modifier 1 in the Part 1 and Prime in the Part 2 column.
4. In the Glossary tab, add First Name and Last Name and the corresponding abbreviations. Select the P check box.
5. Open the Model Naming Options dialog and click the Name Mapping tab.
6. Select the Use Glossary check box for Entity to Table and Attribute to Column rows. In the Prefix column, enter Tbl_ in the first row and Col_ in the third row. Click Close.

   The standards that you have defined are applied to the model.

Import Naming Standards from a Previous Version

If you have created naming standards in a previous version of CA ERwin DM, you can import them to the current release. When you import naming standards, the definitions for logical and physical models, and glossary words are imported.

Import from a Previous Version

If you have created naming standards for a previous version of CA ERwin DM, you can import them to an NSM Option object. You import a .NSM file.

Follow these steps:

1. Expand the Model Explorer, right-click NSM Options and click New.

   A new NSM Option object is added.

2. Right-click the new object (or an existing object) and click Properties.

   NSM Option Editor appears.

3. Click Import Other.

4. Select the .NSM file and then click Open.

   The NSM file is displayed as a new NSM Option object. The standards for Logical and Physical models are displayed in the respective tabs. The glossary words and corresponding abbreviations are displayed in the Glossary tab.

Apply Naming Standards by Binding a Model Template

When you bind a model template to the current model, the naming standards are also applied. If you run the wizard to select individual objects to bind, select the Naming_Options, NSM_Option, and NSM_Glossary_Word check boxes.

Bind a Template to a Model

Binding a model template to a data model copies the non-built-in objects from the template model to the target model. When you bind a model template, the predefined model objects appear in your model. Binding eliminates the need to enter objects manually or depend on reverse engineering. Changes to the model template are reflected in the models that are bound to it, maintaining consistency of object names. Whenever you open a model with a bound model template, you can synchronize any model template changes with the objects used in the model.

Binding of model templates can occur at any time necessary, including during model creation, or after the model is created.

Objects that are brought in to a model from a template fully respect modeling rules. If any conflicts occur in the model, the existing data is preserved as much as possible.

Follow these steps:

1. Click File, Model Templates, Bind Template.

   The Bind Model Template dialog opens.

2. Select the model template to bind to the current model using the drop-down control. Use the toolbar to browse for the template file locally on your computer or one that is stored in a mart.

3. Select one of the following options to specify how to bind the template, and click OK.

   Load the entire contents of the template

   Specifies to load the entire contents of the template. This option binds all of the model template objects to the current model without the opportunity to review and select individual template objects. The NSM Option object that is active in the model template is made active in the current model as well.

   Run the Template Wizard to select objects for synchronization

   Specifies to open the Model Template Synchronization Wizard so that you can select the objects in the model template to synchronize with the data model.

   The model opens in the workspace with the template objects copied into it.

Verify Naming Standards Compliance

To know if the naming standards that you have created are applied properly, use the Naming Standards Compliance dialog.

Follow these steps:

1. Click Tools, Standards, Check Naming Standards Compliance.

   The Check Name Standards Compliance dialog opens.

2. Select the objects to check, select the Check Word Order and Position check boxes, and then click Start.

   All instances of non-compliance are displayed in the Check Name Standards Compliance dialog. You can replace the name, replace all instances of the name, or ignore the non-conforming name.

Apply Naming Standards to a Model

After you define naming standards using the NSM Option Editor, select individual model objects and apply the standards. The glossary words and abbreviations (or alternate abbreviations) that are defined in the active NSM Option object are applied as naming standards to a model.

Follow these steps:

1. Open a model that includes an active NSM Option object.
2. Click Actions, Model Naming Options.

   The Model Naming Options dialog opens.
3. Click the Name Mapping tab.
4. For each model object type that you want to use the glossary, select the Use Glossary check box and select the type of abbreviation. Click OK.

   The naming standards are applied when you switch from a logical model to a physical model.

   In a Logical/Physical model, define the naming standards, create the abbreviations, and select the object that uses the glossary in the Logical view. Switch from the Logical view to the Physical view and the standards are applied.

   In a Logical Only (LO) or a Physical Only (PO) model, define the naming standards, create the abbreviations, select the object that uses the glossary, and then derive a PO or an L/P model. In the Derive Model wizard, go to Naming Standards. Select the model that has an active NSM object as the active model template. The naming standards are applied to the derived model.

Note: The translation from logical to physical through the NSM Option Object works only if the physical name is inherited from the logical name. That is, only if the names have not been manually changed in the physical model. If a physical name is modified in a logical/physical model, the inheritance from the logical side to the physical side is overridden. The naming standards no longer work. However, you can reset the override property to inherit from the logical name to restore this inheritance.

NSM File Attached to an Older Version Model

When you open a model from an older version of the product, it is upgraded to the current release. If an NSM file was attached to the model, the file is not imported and attached to the model automatically. Import the NSM file to a template and attach the template to the model manually.

Example

Use this example to create database specific templates and standardize database specific properties across all your data models. If you want to include the Naming Standards in each database specific target template, you must change the naming standards glossary list for each template. You must change the glossary for each template, because the Naming Standards information is stored within a model. This section describes how you can manage Naming Standards from a single source.

Follow this process:

- Create a logical-only model template with general model set up criteria and include the Naming Standards information.
- [Optional] Create a database specific template and use this template as the binding source and select the “Synchronize on Model Load” option.
- Attach this template as the binding template source for your model and select the Synchronize on Model Load option.

  If you have to change something in the glossary, you just change the Naming Standards object in the logical model template and save the template. When you open a model, if the auto-synchronize option is enabled, the database specific template model is opened. Next, the logical model template is opened and the Naming Standards object is synchronized with the database specific template. The remaining template information is also populated in the new model.

Example

This section includes two examples with two different databases—DB2 and Oracle. The examples use templates that are stored in a Mart.

1. Create a logical model template that includes the following basic standards desired for all data models:
   - A standard theme.
   - An Annotation that enables the model name to be presented on diagrams (a type of legend).
   - Standard Domains.
   - Standard Entity level User Defined Properties (UDP) that can be shared among all data models.
   - A set of start-up diagrams. Each diagram includes a diagram shape to enable the presentation of the diagram name in each diagram.
   - A Naming Standards object.

2. Open this model and save it to a Mart as a template; provide the name "TmplGeneral".

   To create each database specific template in ERwin:

   a. Click File, New.

   b. Select Type as Logical/Physical.

   c. Select the Target Database as Oracle.

   d. Select TmplGeneral for Model template.

   e. Select Preserve the template binding.

   You now have a logical/physical model with all the pre-populated information from the TmplGeneral logical model template. You can add any common DB2 characteristics that you want all your DB2 models to share.

In addition, make the following changes:

1. Click Model, Model Properties and change the model name to TmplDb2.

2. Click Model Explorer, Annotations - Change the Annotation to reflect the model name:

3. Perform the following steps to prevent the over writing of model names during synchronization:

   a. Click File, Model Template, Run the Template Wizard. The Model Template Synchronization Wizard appears.

   b. Go to the Model page and change the action of the Annotation object named "ModelLegend" to Ignore. This setting bypasses synchronizing the model name.

   c. In the Settings page, select the Ignore synchronization of the root model object option. This setting prevents over writing of the model level properties from the logical only model.

   d. Click Finish.

4. Select Actions, Model Naming Options. Verify whether the Use Glossary option in the Name Mapping tab is set for the model objects for which you want to leverage the naming standards glossary.

5. Also, as these diagrams were synchronized from a logical only model, go to the Diagram Properties dialog for the Physical Overview and Relational Key diagrams and change the View Mode to "Physical".

6. Save this model to your mart library as model name "TmplDB2" (with the Mark as Template option)

7. Repeat these steps for each database specific template (i.e. SQL Server, Oracle, Teradata, etc.). With this, you can add any database specific properties in your templates to use across all application data models.

You can now use these database specific templates for your application data models. Either create a new model, or, bind an existing data model to the database specific template model manually. If you are binding an existing model, review the model to verify that no properties in the template model will cause issues with the existing model when synchronized.

In this example, creating a new application data model using the template is used as a starting point.

1. Select File, New.
2. Select Match template.
3. Select the TmplDB2 template model from the Mart.
4. Select Preserve the template binding option.
5. Click OK.

   Now you have a DB2 application data model with all the properties from both the TmplGeneral template and the TmplDB2 template.
6. If you use the Annotation ModelLegend text box, select Ignore in the template. Selecting ignore prevents this object from being updated during synchronization.
7. Save this model to your Mart library. To follow this example, save it as model “AppModelDB2”.
8. To test the synchronization process, add a few Entities in the “Logical Overview” diagram and drag/drop the domain “Identification” as primary keys to both Entities. (Select the Identification domain and drag it to the primary key locations in each Entity).
9. Select the “Physical Overview” diagram. Notice that the word Identification changed to its abbreviation from the Naming Standards Glossary.
10. Save and Close the model as AppModelDB2.
11. Open the original model template TmplGeneral and change the abbreviation for Identification from “Id” to “Idnt”.
12. Save and close this model.
13. Open model AppModelDB2.

    Notice that the abbreviation is automatically applied through the template hierarchy to this application data model for DB2.

Name Hardening Wizard

The Name Hardening Wizard lets you override inheritance of names and block name changes or resets within a model. Using the wizard, you can select the object types and object instances for which you must harden logical or physical names, or both. The wizard helps you override the name inheritance for all instances of the selected object types, a subset of those instances, or a specific instance.

To run the Name Hardening Wizard, click Actions, Manage Name Hardening.

You can harden names for the following object types:

- Attribute/Column
- Default value
- Domain
- Key group/Index
- Relationship
- Entity/Table
- Validation rule

Note: If you prefer, you can select all the objects directly on the diagram, launch the wizard, and go directly to the Action page. The objects you select on the diagram populate the Type Selection and Object Selection pages.

The Name Hardening Wizard provides three options: Type Selection, Object Selection, Action. Using the the three options you can perform the following action:

- Override Name Inheritance (see page 66)
- Unharden Object Names (see page 67)

Override Name Inheritance

You can override the inheritance of names and can harden the names. If a name is hardened, the physical model name does not change upon changing the logical name.

Follow these steps:

1. Click Actions, Manage Name Hardening.

   The Name Hardening Wizard opens.

2. Click Type Selection.

   The options that you select on this page determine the options that are defined on the Object Selection page.

   Note: While you can select the Entity check box in a logical model, you can harden names only in a physical model. Logical entity names are not inherited names; they are set on the entity object. Therefore, they do not change and there is no need to harden them.

3. Select the object types for which you want to override inheritance, then click Object Selection.

4. Select the model objects for which you want to override inheritance, then click Action.

   If some of the subordinate objects are selected, then the check box for that collection node is selected and is also shaded.

5. Select the check box for each item for which you want to override inheritance, and then click Finish.

   The Logical and Physical columns indicate the status of the objects and the Selected Objects column lists their fully qualified names. For example, instead of simply 'a,' an attribute name would be 'OwnerName.E/1.a.'. When you unharden an object name, the same name you used in the hardening process remains. However, now you can edit or reset that name (editing or resetting of hardened names is not allowed).

   Note: Unhardening an object name does not automatically restore inheritance, it only allows the names to be reset or changed.

Unharden Object Names

The Name Hardening wizard can be used to unharden names. When you unharden an object name, the same name you used in the hardening process remains. After the unharden process is complete, you can edit or reset the name (editing or resetting of hardened names is not allowed).

Note: Unhardening an object name will not automatically restore inheritance, it only allows the names to be reset or changed.

Follow these steps:

1. Click Actions, Manage Name Hardening.

   The Name Hardening Wizard opens.

2. Click Type Selection.
3. Select the object types that you want to unharden, then click Object Selection.
4. Select the model objects that you want to unharden, then click Action.
5. Select the check box for each item for which you want to unharden the names, then click Finish.

   The name for each object that you selected is unhardened, and the Name Hardening Wizard closes.

Applying Data Type Standards

A data type is a predefined set of characteristics for an attribute or column that specifies field length, acceptable characters, and optional and required parameters. For example, the data type Char(18) specifies that the column can store up to 18 alpha and numeric characters.

By default, a data type is applied to every attribute in a logical model and to every column in a physical model. In the logical model, you assign the data type or the domain from which the attribute inherits its properties determines it. In the physical model, you assign the data type or the default value that is specified by the target server determines it.

Considering the large quantity of attributes or columns a data model usually has, it is tedious to assign and maintain consistency of data types manually. The DSM Option Editor helps you easily assign and maintain data types in a model.

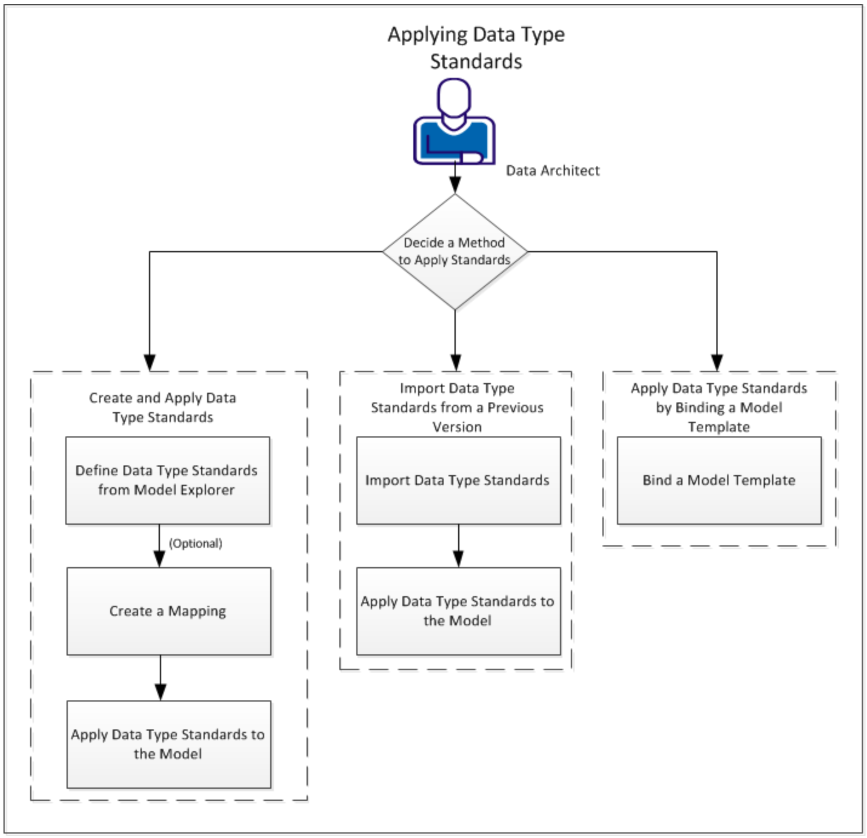
Create data type standards so that everyone in your organization uses data types consistently.

When you create a DSM Option object, it is preloaded with default logical data types. Use the DSM Options Editor to create data types and mappings. You can create multiple DSM Option objects but you can apply only one of them to the model.

You can assign data types in any of the following scenarios:

- Deriving a physical model from a logical model, or switching from a logical model to a physical model. A physical data type is automatically assigned to each column based on the data type assigned to the corresponding attribute.
- Deriving a physical model from a physical model. The corresponding data type for the target database is automatically assigned.
- Migrating from one version of a database to a newer version, or changing the target database. You are prompted you to verify if you want to convert data types. When you convert the data types, the corresponding data type for the target database is automatically assigned.
- Creating a physical model by reverse engineering from a database or script file. Information from the database or script file is extracted, and a physical data type is automatically assigned to each column.

You can apply data type standards in three ways, as illustrated in the following diagram:


Applying Data Type Standards
Data Architect
Decide a Method to Apply Standards
Create and Apply Data Type Standards
Define Data Type Standards from Model Explorer
(Optional)
Create a Mapping
Apply Data Type Standards to the Model
Import Data Type Standards from a Previous Version
Import Data Type Standards
Apply Data Type Standards to the Model
Apply Data Type Standards by Binding a Model Template
Bind a Model Template

To apply data type standards, follow these steps:

Decide a method to apply standards (see page 69).

To create and apply data type standards, follow these steps:

1. Define data type standards from Model Explorer (see page 70).
2. (Optional) Create a mapping (see page 71).
3. Apply data type standards to the model (see page 71).

To import data type standards from a previous version, follow these steps:

1. Import data type standards (see page 72).
2. Apply data type standards to the model (see page 72).

Apply data type standards by binding a model template:

1. Bind a model template (see page 73).

Decide a Method to Apply Standards

You can apply data type standards in three ways:

Create a standard

Create a data type standard and apply it to your model if you are applying standards for the first time. After you create a DSM object and define your standards, you can use the standard in all your data models.

Import from a previous release

Import data type standards from a previous release and apply the standard to your model. Importing saves your time from creating a standard and maintains the consistency across all data models in your organization.

Bind a model template

Bind an existing model template having data type standards to your model. You can use the same template for multiple models. You can also sync the template with the current model for any changes in the model. After you sync the template, it will inherit the changes that are made in the current model.

Create and Apply Data Type Standards

Create data type standards if you are implementing standards for the first time.

Define Data Type Standards from Model Explorer

Define data type standards to maintain uniform data type standards across all data models.

Follow these steps:

1. Expand Model Explorer, right-click DSM Options, and click New.

   A data type standards object is added.

2. Right-click the newly created object and click Properties.

   The DSM Option Editor opens.

3. (Optional) Rename the DSM object in the Name field.
4. Click the Logical Definition tab and edit the Domain, Length, Precision, and Default Length values for the required data type.
5. (Optional) To create a data type, click New under the Logical Definition tab.

   A data type is created.

6. (Optional) Rename the new data type and change the properties.

   A data type standard object is defined and saved.

Note: To delete a DSM object or a data type, select the DSM object or the data type row and click Delete.

If you want to create a mapping, you do not have to close the DSM Option Editor window.

(Optional) Create a Mapping

Map data types between different databases to maintain uniform data types in all the data models. Mapping lets you eliminate the differences in the data types when you import a model from one database to another. Also, mapping enables you to use new data types in physical models.

Note: You can create multiple DSM Datatype Mapping objects and DSM Datatype Mapping entries in the DSM Datatype Mapping Editor.

Follow these steps:

1. In the DSM Option Editor, select the Mapping tab and click New.

   A DSM Datatype Mapping object is created.

2. Click Edit.

   The DSM Datatype Mapping Editor opens.

3. (Optional) Edit the name of the DSM Datatype Mapping object.
4. Select the Mapping tab, and select the source database from the From drop-down list and the target database from the To drop-down list.
5. (Optional) Edit the To Datatype field.
6. (Optional) Add a data type for mapping.
7. Click Close.

   The DSM Datatype Mapping Editor closes and you return to the DSM Option Editor dialog.

Apply Data Type Standards to the Model

After you define data type standards, apply the standards by attaching the DSM object to the model. If you have created multiple DSM object, you can attach only one DSM object to the model.

To attach a DSM object to a model, open the DSM Option Editor from Model Explorer, and select the Is Active check box.

Import Data Type Standards from a Previous Version

If you have created data type standards in a previous version of CA ERwin DM, you can import it to CA ERwin DM version 9. When you import data type standards, mappings are also imported.

Import Data Type Standards

If you have upgraded to the latest release of the r9 series, you can import the data type standards from a previous version. Importing the standards maintains consistency across all data models in your organization and saves time from creating standards.

Note: The file that you are importing must have the .DSM extension.

Follow these steps:

1. Expand the Model Explorer, right-click DSM Options and select New.

   A new DSM Option object is added.

2. Right-click the new object (or an existing object) and select Properties.

   The DSM Option Editor appears.

3. Click Import Other.
4. Select the .DSM file from the stored location and click Open.

   The DSM file is displayed as a new DSM Option object.

Apply Data Type Standards to the Model

After you import the data type standards object, apply the standards by attaching the DSM object to the data model.

To attach a DSM object to a model, open the DSM Option Editor from Model Explorer, and select the Is Active check box.

Apply Data Type Standards by Binding a Model Template

When you bind a model template having data type standards to the current model, data type standards are also applied.

Bind a Model Template

Binding a model template to a data model copies the non-built-in objects from the template model to the target model. When you bind a model template, the predefined model objects populate your model. Therefore, binding eliminates the need to enter objects manually or depend on reverse engineering. Changes to the model template are displayed in the models that use it, maintaining consistency of object names. Whenever you open a model with a bound model template, you can synchronize any model template changes with the objects used in the model.

Binding of model templates can occur at any time necessary, including during model creation or after the model is created.

The objects that are brought in to a model from a template fully respect modeling rules. If any conflicts occur in the model, the existing data is preserved as much as possible.

Follow these steps:

1. Click File, Model Templates, Bind Template.

   The Bind Model Template dialog opens.

2. Use the toolbar to browse for the template file locally on your computer or in Mart. If you have used the bind template feature before, use the drop-down control to select the model template to bind to the current model.

3. Select one of the following options to specify how to bind the template, and click OK.

   Load the entire contents of the template

   Specifies to load the entire contents of the template. This option binds all of the model template objects to the current model without the opportunity to review and select individual template objects.

   Run the Template Wizard to select objects for synchronization

   Specifies to open the Model Template Synchronization Wizard so that you can select the objects in the model template to synchronize with the data model.

   The model opens in the workspace with the template objects copied into it. The data type standard is applied on the model.

Appendix A: Supported Metadata Integration Bridges

Table of Supported Bridges

The following table lists the Meta Integration Technology, Inc. (8.0.2) bridges that are currently available in CA ERwin Data Modeler (as of Release r9.6):

Note: Read the information displayed in the tip text area on the Source and Destination pages of the Import from External Format and Export to External Format dialogs carefully. Complete any requirements as appropriate before proceeding.

Application	Mode
Altova XMLSpy	Export
Apache Hadoop Hive Native bridges include: Apache Hadoop Hive Server Apache Hadoop HCatalog Cloudera Enterprise (Hadoop Hive) Cloudera Impala (Hadoop Hive) DataStax Enterprise (Hadoop Hive) Hadapt Hadoop Hive Server Hortonworks Hadoop Hive MapR Hadoop Hive	Import
Borland Together (via UML 1.x XMI)	Import/Export
Borland Together (via UML 2.x XMI)	Import
CA Component Modeler 3.52 (ParadigmPlus)	Import/Export
CA Component Modeler 4.x (via UML 1.x XMI)	Import/Export
CA COOL:Biz 5.1	Import
CA COOL:BusinessTeam (GroundWorks) 2.2.1	Import
CA COOL:DBA (Terrain for DB2) 5.3.2	Import
CA COOL:Enterprise (ADW) 2.7	Import
CA COOL:Xtras Mapper (TerrainMap for DB2)	Import
CA ERwin 3.x (ERX)	Import
CA ERwin 4.x Data Modeler	Import

Application	Mode
CA ERwin Data Profiler - Beta bridge	Import
CA ERwin Web Portal	Export
CA Gen	Import/Export
COBOL Copybook Flat Files	Import
Embarcadero Describe (via UML 1.x XMI)	Import/Export
Embarcadero ER/Studio Business Architect (via DM1)	Import/Export
Embarcadero ER/Studio Data Architect	Import/Export
Embarcadero ER/Studio Repository	Import
EMC ProActivity 3.x & 4.0	Import
Gentleware Poseidon (via UML 1.x XMI)	Import/Export
Grandite Silverrun-RDM 2.4.4 to 2.8.6	Import
Google BigQuery	Import
IBM Cognos Framework Manager (FM)	Import/Export
IBM Cognos Framework Manager (FM) for ReportNet	Import/Export
IBM Cognos BI Reporting - Framework Manager	Export
IBM Cognos Content Manager Packages	Import
IBM DB2 Cube Views	Import/Export
IBM DB2 Data Server (via JDBC)	Import
IBM InfoSphere Data Architect (IDA)	Import/Export
IBM InfoSphere DataStage	Import/Export
IBM InfoSphere Discovery - Beta bridge	Import
IBM InfoSphere Metadata Workbench Extension Mappings - Beta	Export
IBM InfoSphere Warehouse - InfoSphere Data Architect	Import/Export
IBM Netezza Database (via JDBC)	Import
IBM Rational Rose 4.0	Import/Export
IBM Rational Rose 6.0(98i) to 6.5(2000)	Import/Export
IBM Rational Rose 7.x (2000e and newer)	Import/Export
IBM Rational Rose XDE Developer (via Rose MDL)	Export
IBM Rational Software Architect (RSA) (via Rose MDL)	Export
IBM Rational Software Architect (RSA) (via UML 2.x XMI)	Import

Application	Mode
IBM Rational Software Modeler (RSM) (via Rose MDL)	Export
IBM Rational System Architect (SA) 10.4 to 11.x (Encyclopedia)	Import/Export
IBM Rational System Architect (SA) 7.1 to 11.x (File)	Import/Export
IBM Telelogic Tau (via UML 1.x XMI)	Import/Export
Informatica Data Analyzer	Export
Informatica Developer	Import/Export
Informatica PowerCenter	Export
Informatica PowerCenter (File)	Import
Informatica PowerCenter (Repository)	Import
IRI CoSORT RowGen Data Definition File	Export
IRI CoSORT SortCL Data Definition File	Export
Micro Focus (Merant) AppMaster Builder	Import
Microsoft Office Excel	Import/Export
Microsoft Office Visio Database (via ERX)	Import
Microsoft Office Visio UML (via UML 1.x XMI)	Import
Microsoft SQL Server Analysis Services (Repository)	Export
Microsoft SQL Server Analysis Services (via DSV File)	Import/Export
Microsoft SQL Server Data Source View	Import/Export
Microsoft SQL Server Integration Services (File)	Import/Export
Microsoft SQL Server Integration Services (via DSV File)	Import/Export
Microsoft SQL Server XML Data Reduced (XDR) Schema	Import
Microsoft SQL Server Analysis Services 2005 to 2012 (Repository)	Import
Microsoft SQL Server Database (via JDBC)	Import
Microsoft SQL Server Integration Services (Repository)	Import
Microsoft Visual Studio / Modeler 2.0 (via MDL)	Import/Export
MicroStrategy 7.0 to 9.x	Import
NoMagic MagicDraw (via UML 1.x XMI)	Import/Export
OMG CWM 1.x XMI 1.x	Import/Export
OMG CWM Pre-1.0 XMI 1.1	Import/Export
OMG UML 1.x XMI 1.x	Import/Export

Application	Mode
OMG UML 2.x XMI 2.x	Import
Oracle Business Intelligence (OBI) Server	Import
Oracle Database (via JDBC)	Import
Oracle Data Integrator (ODI)	Import/Export
Oracle Data Warehouse Administrative Console (DAC)	Import
Oracle Designer	Import/Export
Oracle Hyperion Essbase Integration Services (EIS)	Import/Export
Oracle Warehouse Builder (OWB)	Import/Export
Pivotal Greenplum Database (via JDBC)	Import
PostgreSQL Database (via JDBC)	Import
SAP BusinessObjects Data Integrator (BODI)	Import/Export
SAP BusinessObjects Data Integrator (BODI)	Import
SAP BusinessObjects Designer (File)	Import/Export
SAP BusinessObjects Designer (Repository)	Import
SAP BusinessObjects Information Design Tool (IDT) (Repository)	Import
SAP BusinessObjects Information Design Tool (IDT) (File)	Import/Export
SAP NetWeaver Master Data Management (MDM) - Beta bridge	Import/Export
Select SE	Import
Sparx Enterprise Architect (EA) (via UML 1.x XMI)	Import/Export
Sparx Enterprise Architect (EA) (via UML 2.x XMI)	Import
Sybase ASE Database (via JDBC)	Import
Sybase PowerDesigner CDM 6.x	Import
Sybase PowerDesigner CDM 7.5 to 16.x	Import
Sybase PowerDesigner CDM 8.0 to 16.x	Export
Sybase PowerDesigner OOM 9.x to 15.x (via UML 1.x XMI)	Import/Export
Sybase PowerDesigner OOM 9.x to 15.x (via UML 2.x XMI)	Import
Sybase PowerDesigner PDM 6.1.x	Import
Sybase PowerDesigner PDM 7.5 to 16.x	Import
Sybase PowerDesigner PDM 8.x to 16.x	Export
Sybase PowerDesigner LDM 15.x to 16.x	Import

Application	Mode
Sybase PowerDesigner XSM 10.x to 15.x	Export
Syncsort DMExpress	Import/Export
Tableau - File	Import
Tableau Server (Repository)	Import
Talend Data Integration - Beta bridge	Import/Export
Teradata Database (via JDBC)	Import
Tigris ArgoUML (via UML 1.x XMI)	Import/Export
Visible IE:Advantage 6.1	Import
W3C XML DTD 1.0	Import
W3C XML Schema 1.0 (XSD)	Import/Export

Notes:

- For information about how to use the metadata integration wizards, see the CA ERwin Data Modeler online help.
- For Oracle Data Integrator 11g, the new Java API based architecture is supported.
- For Oracle Warehouse Builder 11.2, the new API architecture for 11.2 (different from 11.1) is supported.
- IBM Rational System Architect 7.1 to 11.x (File) and IBM Rational System Architect 10.4 to 11.x (Encyclopedia) must be installed on your local computer before you use these import and export bridges.

Index