

erwin[®] Data Modeler

erwin Metamodel Overview

Release 2018 R1

This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the "Documentation"), is for your informational purposes only and is subject to change or withdrawal by erwin Inc. at any time. This Documentation is proprietary information of erwin Inc. and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of erwin Inc.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all erwin Inc. copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to erwin Inc. that all copies and partial copies of the Documentation have been returned to erwin Inc. or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, ERWIN INC. PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL ERWIN INC. BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF ERWIN INC. IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is erwin Inc.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2018 erwin Inc. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

erwin Product References

This document references the following erwin Inc. products:

- erwin® Data Modeler (erwin DM)

Contact erwin

Understanding your Support

Review [support maintenance programs and offerings](#).

Registering for Support

Access the [erwin support](#) site and click Sign in to register for product support.

Accessing Technical Support

For your convenience, erwin provides easy access to "One Stop" support for all editions of [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
- erwin Support policies and guidelines
- Other helpful resources appropriate for your product

For information about other erwin products, visit <http://erwin.com/products>.

Provide Feedback

If you have comments or questions, or feedback about erwin product documentation, you can send a message to techpubs@erwin.com.

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.

Contents

Chapter 1: The erwin Data Modeler Metamodel **7**

Introduction7

Uses7

Related Documentation8

Chapter 2: Diagrams **9**

Entity and View Diagram9

Model and Subject Area Diagram10

Entity, Attribute, and Key_Group Diagram11

Entity, Attribute, Key_Group, and Relationship Diagram12

Attribute, Domain, Default, and Validation Rule Diagram13

Chapter 1: The erwin Data Modeler Metamodel

This section contains the following topics

[Introduction](#) (see page 7)

[Uses](#) (see page 7)

[Related Documentation](#) (see page 8)

Introduction

Metadata is data about data. Data Architects use model metadata to describe the business information and database structures depicted by the diagrams developed and maintained using erwin® Data Modeler. Model metadata typically includes objects such as ENTITY, TABLE, ATTRIBUTE, COLUMN, KEY GROUP, and INDEX. Model metadata also includes properties such as Entity Name, Attribute Definition, Column Data Type, Key Group Type, and Index Owner.

Metadata information has been available in erwin® Data Modeler for years using XML export/import and the SCAPI API. An additional mechanism was added to facilitate access to underlying model metadata further, and to assist Data Architects to use this information effectively.

Uses

An ODBC connectivity interface is provided that permits SQL-based queries to retrieve model metadata from any open model. Using any ODBC-compliant reporting product, you can retrieve:

- Information related to model objects and properties
- The metadata defining those objects and properties
- The action log that records the changes that were made to the model during the current modeling session

You can also edit the SQL templates used by the Forward Engineering and Alter Script processes to produce SQL customized exactly to your specifications.

The HTML document describes all of the accessible metadata associated with erwin® Data Modeler, and the relationships that exist between the individual metadata components.

Related Documentation

The ODBC connectivity layer and reporting interface is documented in the *Creating a Report on the Model Metadata* guide. The *Creating a Report on the Model Metadata* guide is installed to the \erwin Bookshelf\Bookshelf_Files\HTML\ODBC Reporting directory.

Chapter 2: Diagrams

This section includes diagrams that illustrate some of the key portions of the metamodel. These diagrams are provided as a visual counterpart to the textual descriptions of the metadata.

The diagrams are presented in pairs, one using UML notation and the other using IDEF1X notation. The UML notation presents a raw view of the metamodel; the IDEF1X notation presents the view exposed using the ODBC interface.

Note: Some of the diagrams include abstract object types. Abstract object types are colored light blue in the diagrams. Although the definitions of these object types exist in the metamodel (termed 'M1'), no instances of them exist within any data model (termed 'M0'). Because no instances exist in any data model, abstract object types are unavailable for M0 queries using the ODBC interface.

Some of the IDEF1X diagrams show vector properties (properties with more than one value). Vector properties are shown as tables colored orange. For more information about the transformations between the raw metamodel view and the ODBC metamodel view, see the *erwin® Data Modeler ODBC Reporting Guide*.

This section contains the following topics

[Entity and View Diagram](#) (see page 9)

[Model and Subject Area Diagram](#) (see page 10)

[Entity, Attribute, and Key Group Diagram](#) (see page 11)

[Entity, Attribute, Key Group, and Relationship Diagram](#) (see page 12)

[Attribute, Domain, Default, and Validation Rule Diagram](#) (see page 13)

Entity and View Diagram

This diagram shows the object type structure for Entities, Views, and Cached Views. This diagram makes the following assertions:

- *Entity* is a type of *Abstract_Entity*
- *Abstract_View* is a type of *Abstract_Entity*
- *Cached_View* is a type of *Abstract_Entity*
- *View* is a type of *Abstract_Entity*

Entity and View Diagram--UML Notation:

Entity and View Diagram--IDEF1X Notation:

Model and Subject Area Diagram

This diagram shows the storage of Subject Area membership information and makes the following assertions:

- *Abstract_ERwin_Object* is a type of *Abstract_Object*
- The object types *Abstract_Model_Node*, *Model*, *Subject_Area*, and *Relationship* are all types of *Abstract_Objects*
- The object types *Subtype_Symbol* and *Abstract_Entity* are both types of *Abstract_Model_Nodes*
- *Abstract_Model_Nodes* are owned by *Model*
- *Relationships* are owned by *Model*
- *Subject_Areas* are owned by *Model*
- *Subject_Areas* hold their collection of member *Abstract_Model_Nodes* in *Referenced_Entities_Ref*. When viewed using SCAPI, this is held as a vector reference property on the *Subject_Area*. When viewed using ODBC, the associative table *Referenced_Entities_Ref* stores this membership information.
- *Subject_Areas* hold their collection of member *Relationships* in *Referenced_Relationships_Ref*. When viewed using SCAPI this is held as a vector reference property on the *Subject_Area*. When viewed using ODBC, the associative table *Referenced_Relationships_Ref* stores the membership information.

Note: This data is maintained by erwin® Data Modeler based on the data in *Referenced_Entities_Ref*. Do not attempt to directly change this data.

Model and Subject Area Diagram--UML Notation:

Model and Subject Area Diagram--IDEF1X Notation:

Entity, Attribute, and Key_Group Diagram

This diagram shows the structure of Entities and Attributes and their associated Key_Groups and Key_Group_Members. This diagram makes the following assertions:

- *Attributes* are owned by *Entities*
- *Key_Groups* are owned by *Entities*
- *Key_Group_Members* are owned by *Key_Groups*
- *Key_Group_Members* are linked to their associated *Attribute* using the *Attribute_Ref* property
- *Key_Group_Members* are linked to their parent *Key_Group_Member* using the *Parent_Key_Group_Member_Ref* property
- *Attributes* are linked to their parent *Attribute* using the *Parent_Attribute_Ref* property

Entity, Attribute, and Key_Group Diagram--UML Notation:

Entity, Attribute, and Key_Group Diagram--IDEF1X Notation:

Entity, Attribute, Key_Group, and Relationship Diagram

This diagram extends the prior diagram by adding Relationships. The following assertions are added:

- *Key_Groups* reference their associated *Relationships* using the *Key_Group_Ref* property
- *Relationships* reference their associated *Key_Group* using the *Key_Group_Ref* property
- *Relationships* reference their parent *Entity* using the *Parent_Entity_Ref* property
- *Relationships* reference their child *Entity* using the *Child_Entity_Ref* property
- The collection of *Relationships* in which an *Entity* participates as the parent is stored in the *Parent_Relationships_Ref* property. When viewed using SCAPI, this is held as a vector reference property on *Entity*. When viewed using ODBC, the associative table *Parent_Relationships_Ref* stores this information.
- The collection of *Relationships* in which an *Entity* participates as the child is stored in the *Child_Relationships_Ref* property. When viewed using SCAPI, this is held as a vector reference property on *Entity*. When viewed using ODBC, the associative table *Child_Relationships_Ref* stores this information.
- *Attributes* reference their parent *Attribute* using the *Parent_Relationship_Ref* property.

Entity, Attribute, Key_Group, and Relationship Diagram--UML Notation:

Entity, Attribute, Key_Group, and Relationship Diagram--IDEF1X Notation:

Attribute, Domain, Default, and Validation Rule Diagram

This diagram shows the links between Attributes, Domains, Defaults, and Validation_Rules. This diagram makes the following assertions:

- *Attributes* reference their parent *Attribute* using the *Parent_Attribute_Ref* property
- *Attributes* reference their parent *Domain* using the *Parent_Domain_Ref* property
- *Domains* reference their parent *Domain* using the *Parent_Domain_Ref* property
- *Default_Constraint_Usage* objects can be owned by *Attributes* or *Domains*
- *Default_Constraint_Usage* objects reference their parent using the *Parent_Default_Constraint_Usage_Ref* property
- *Default_Constraint_Usage* objects reference their associated *Default* object using the *Default_Ref* property
- *Check_Constraint_Usage* objects can be owned by *Attributes* or *Domains*
- *Check_Constraint_Usage* objects reference their parent using the *Parent_Check_Constraint_Usage_Ref* property
- *Check_Constraint_Usage* objects reference their associated *Validation_Rule* object using the *Validation_Rule_Ref* property

Attribute, Domain, Default, and Validation Rule Diagram--UML Notation:

Attribute, Domain, Default, and Validation Rule Diagram--IDEF1X Notation: