

Evaluation guide

Note:

Before using this information and the product it supports, read the information in “Notices” on page 31.

This edition applies to Version 7.5 of IBM Rational Data Architect.

© **Copyright International Business Machines Corporation 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Evaluation criteria for Rational Data Architect	1
Open standards Eclipse integration	1
Integration with other tools	3
Team collaboration	5
Enterprise data modeling	10
Enforcement of enterprise standards	15
Transforming data models	20
Data and information integration	21
Managing data and database objects.	24
IBM Rational Data Architect user roles	27
Bibliography	29
Notices	31
Trademarks and service marks	32

Evaluation criteria for Rational Data Architect

IBM® Rational® Data Architect is a comprehensive Eclipse-based data design and modeling product. Users with different roles in the data design process can use Rational Data Architect alone, or with other IBM products, to model and integrate data in end-to-end data design and development scenarios.

Rational Data Architect key features include:

“Open standards Eclipse integration”

Eclipse-based integrated development environment

“Integration with other tools” on page 3

Communicate and collaborate with other Eclipse tooling and vendor software

“Enterprise data modeling” on page 10

- Create and manage logical and physical data models, with support for heterogeneous databases
- Visualize and edit data models by using data diagrams

“Transforming data models” on page 20

Transform data models to multiple formats

“Enforcement of enterprise standards” on page 15

- Enforce data type standards using domain models
- Enforce naming standards using glossary models
- Enforce additional constraints using data model analysis

“Team collaboration” on page 5

- Share data projects using source code management tools
- Partition data models
- Compare and synchronize data models and data model objects
- Generate data model reports
- Publish data models to a Web
- Save and print data diagrams

“Data and information integration” on page 21

- Discover and relate data sources using information integration modeling
- Browse remote data sources that are defined to a federated database
- Visualize relationships between enterprise objects

“Managing data and database objects” on page 24

- Generate DDL scripts from data objects or data model objects
- Generate delta DDL scripts during compare and synchronization
- Collaborate better between data design and administration

Open standards Eclipse integration

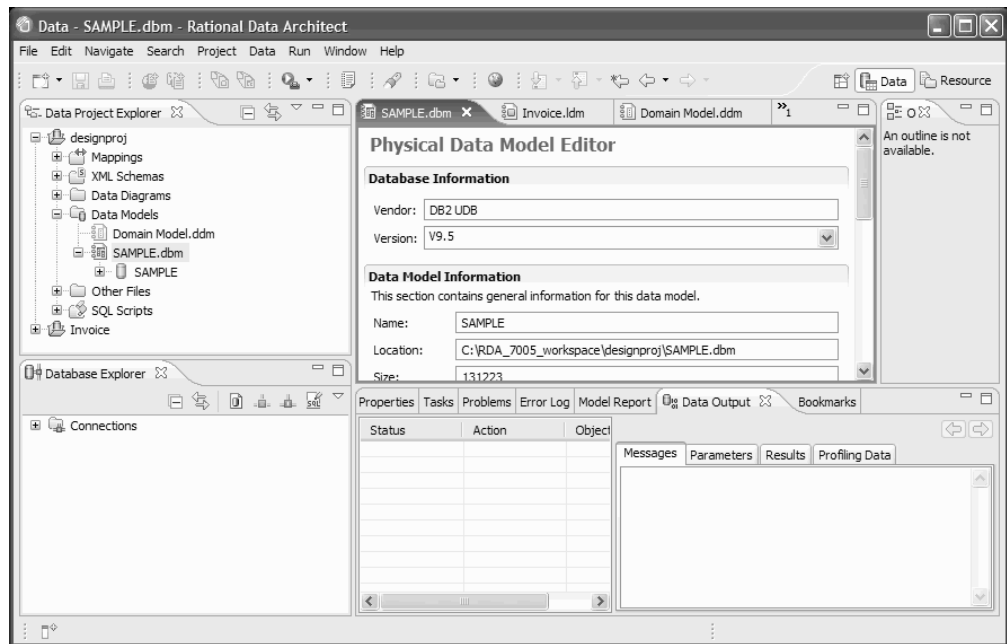
Rational Data Architect is based on the Eclipse platform, an open source integrated development environment (IDE). You can use the integrated Eclipse functionality that is included with Rational Data Architect, and you can install Rational Data Architect into the same Eclipse environment as other compatible Eclipse-based tools.

Eclipse is an open source community whose projects are focused on building an open development platform that is made up of extensible frameworks, tools, and run times for building, deploying, and managing software across the lifecycle. The Eclipse environment is extended, complemented, and supported by major technology vendors, start-ups, universities, research institutions, and individuals.

Rational Data Architect is comprised of the base Eclipse SDK plus the Rational Data Architect functionality that is delivered in a set of Eclipse plug-ins. This means that Rational Data Architect contains all of the base functionality in Eclipse, including source code control and Java™ development features. The plug-in architecture also allows seamless integration with other compatible Eclipse-based tools.

The Eclipse IDE is sometimes called "the workbench". The term *workbench* refers to the desktop development environment. The workbench aims to achieve seamless tool integration and controlled openness by providing a common paradigm for creating, managing, and navigating resources. Each workbench window contains one or more perspectives. Perspectives contain default collections of views and editors and control what appears in certain menus and tool bars. Multiple workbench windows can exist on the desktop at any time.

You can customize the IDE by opening and closing views, adding or removing views from a perspective, or creating custom perspectives. Most of the work that you do in Rational Data Architect is done in the Data Perspective, which includes several dedicated explorer views, editors, and a properties browser.



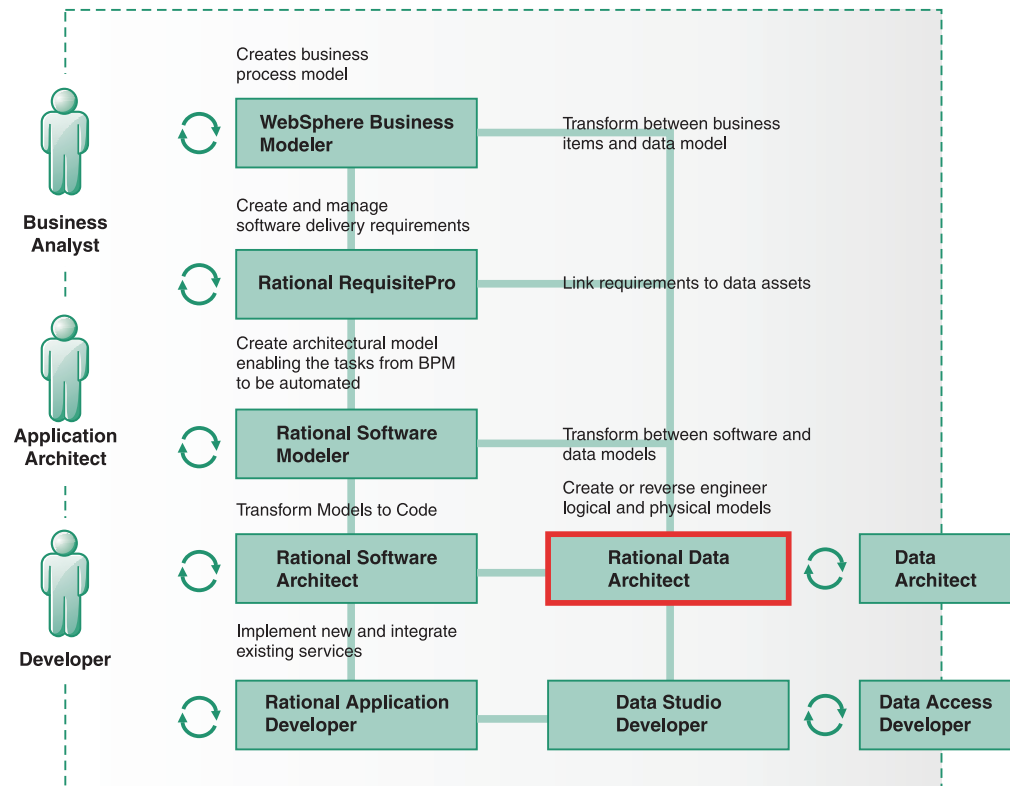
When you have other compatible Eclipse-based tools installed into the same Eclipse instance, you can use their additional perspectives, views, and editors from the same workbench environment, allowing for truly seamless integration.

When you are working in an Eclipse environment, you select a workspace on your local file system. You can then create a data design project in which to store all of your files and artifacts. Data projects can be shared among team members either by using Eclipse Concurrent Versions System (CVS), or by using Rational

ClearCase®. CVS is included with Rational Data Architect and you can easily integrate Rational ClearCase with Rational Data Architect into the same Eclipse environment.

Integration with other tools

You can install Rational Data Architect into the same Eclipse environment as other compatible Eclipse-based tools to use their functionality seamlessly, or you can use the import and export functions in Rational Data Architect to integrate Rational Data Architect with other tools.



Integration with WebSphere® Business Modeler

If you have WebSphere Business Modeler, you can use the XSD transformation process to import WebSphere Business Modeler Business Items that were exported as an XSD into a Rational Data Architect logical data model, or vice versa. Using this feature, the logical data model contains the common semantic view of the information and can be used to generate any number of XML views over the logical model.

Integration with Rational RequisitePro®

You can integrate Rational Data Architect into the same Eclipse environment with IBM Rational Requisite® Pro. IBM Rational RequisitePro integration provides capabilities for associating data model elements with requirements that are identified by a business analyst. You can use the RequisitePro integration to create, manage, and track these project requirements to enable traceability from the database object back up to the requirement and facilitate alignment across roles in the organization.

Integration with IBM Optim

Data masking is used in non-production environments for purposes such as software development and testing, software user training, or data mining and research. You can use Rational Data Architect with the IBM Optim product to generate masked test data from the production data that is contained in a database. You work in a physical data model to complete this task.

You can also generate data model reports that can be shared with other team members (such as security officers, business analysts, architects, database administrators, and quality engineers) to demonstrate regulatory compliance.

In a typical scenario, you use Rational Data Architect to perform the following tasks:

- Annotate atomic domain model objects with data privacy information such as classification, privacy policy, and masking algorithms.
- Specify annotated domain data types for columns during physical data modeling.
- Export annotated physical data models to an Optim export file.
- Open the Optim export file in IBM Optim and use it to create a script that generates masked test data and inserts it into a database.
- Generate a report that demonstrates data privacy compliance.

Integration with other IBM Eclipse-based products

If you have other Rational Software Delivery Platform products, you can use model-driven development with UML to create well-architected applications and services, or use the enhanced software development environment in Rational Application Developer. Rational Data Architect provides bi-directional transformation between the application model expressed in UML and the data model expressed in Information Engineering (IE) notation. You can integrate Rational Data Architect and Rational Software Delivery Platform products into the same Eclipse instance for seamless integration of the functions in each product.

Rational Data Architect is compatible with these Rational Software Platform products:

- Rational Software Architect
- Rational Software Modeler
- Rational Application Developer

In addition, Rational Data Architect is also compatible with Data Studio products:

- The Data Studio Developer product allows database professionals to perform database design, development, deployment, and management tasks.
- The Data Studio Administrator product provides a seamless change management environment.

Integration with the metadata repository of IBM InfoSphere Information Server

If you set up an IBM InfoSphere Information Server metadata repository, you can import and export metadata from Rational Data Architect to the repository. You can combine the business semantics of the IBM WebSphere Business Glossary with the modeling capability of Rational Data Architect to create standardized data models

that comply with your business rules. You can then transform the data models into physical models, and use the physical models to create jobs in the Designer client. You can perform the following tasks:

- Import categories and terms from the repository into Rational Data Architect to create glossary models and optionally transform the glossary model to a logical model during the import process.
- Export metadata from glossary models in Rational Data Architect and import them into the repository to create categories and terms.
- Export metadata from physical models in Rational Data Architect and import them into the repository.
- Transform a domain model or logical model to a glossary model when you are exporting metadata to the repository.

Integration with other modeling tools

You can use Rational Data Architect to import or export data models that were created in other products, such as:

- CA ERwin
- IBM Rational Rose®
- Sybase PowerDesigner
- Telelogic
- Cognos

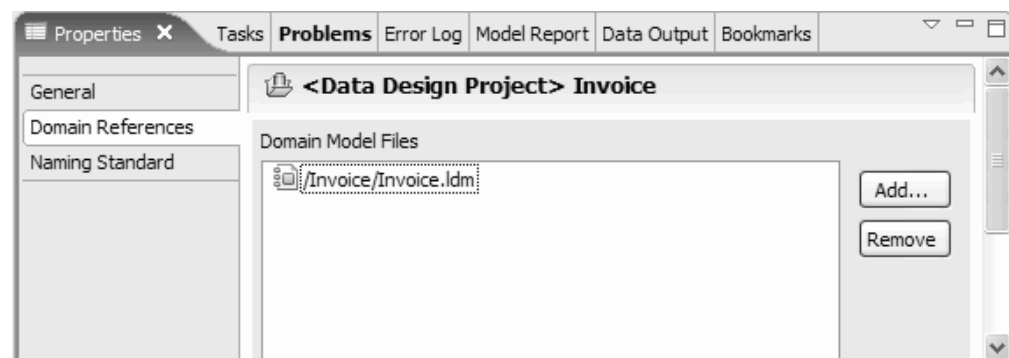
Team collaboration

Rational Data Architect has several features that help team members to collaborate together more effectively.

Data project sharing

Data models are stored in data design projects in Rational Data Architect. You can use either Eclipse Concurrent Versions System (CVS) or Rational ClearCase to share these projects so that team members can work on them concurrently.

You can share glossary models and domain models across data projects so that multiple team members can use the same models. You create an association between the domain and glossary models and the project through the project properties settings. This association ensures that everyone is following the same standards and the glossary and domain models are always in synch.



Data model partitioning

Within a data design project, you can partition logical data model packages into sub-models to facilitate team collaboration and merging. The benefits of data model partitioning include more parallel work across team members, more flexibility, and better partitioning of the logical data model in a team environment so that you can avoid having to merge changes. After you create a sub-model from a package, you can open and access the sub-model through a shortcut in the main model, or you can open and work with a sub-model directly. Containment is maintained between the main model and the sub-models so that logical-to-physical data model transformation, impact analysis, and Web publishing can be run transparently from the main model.

Comparison and synchronization

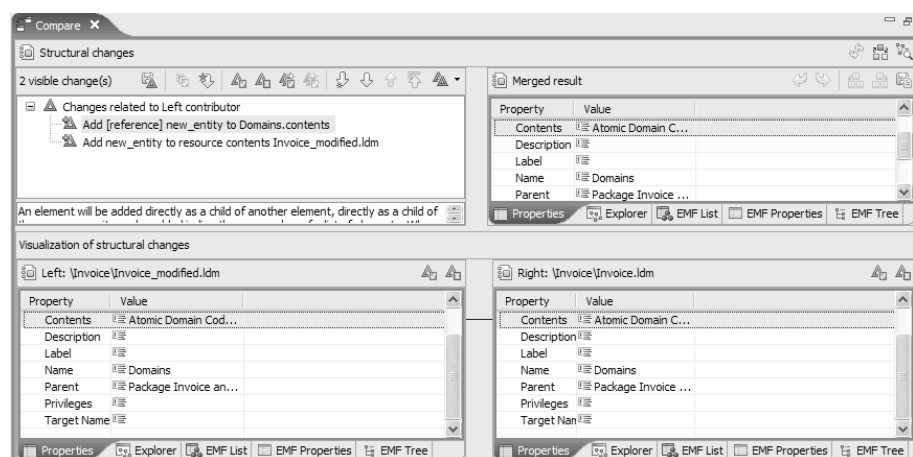
There is an integrated comparison editor in Rational Data Architect that you can use to compare and merge data models or data objects such as schemas and tables - either contained in a data model or in a database. Models or model objects are merged at a semantic level.

If you are comparing data objects that are contained in a data model (for example, two tables), or two objects that are displayed in the Data Source Explorer, you visualize the differences between them in the data object compare editor. You can compare two objects in this way. You can navigate and merge structural differences between data objects, merge properties changes between data objects, generate DDL for changes that you made, and export the structural differences to an XML file on the file system.

If you are comparing data model files (for example, .LDM or .DBM files), you visualize the differences in the data model compare editor. This type of model comparison can be either a two-way or a three-way comparison. For example:

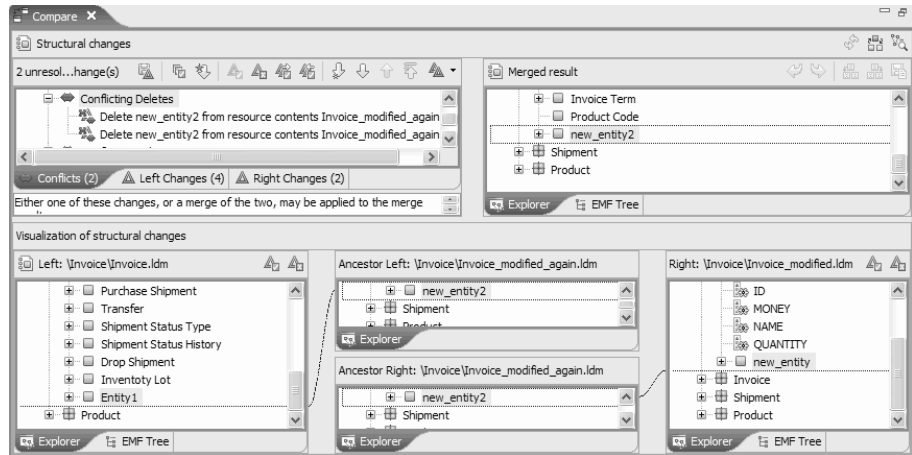
Two-way comparison

The data model compare editor performs a two way comparison if you compare two data models that are not under source code control.



Three-way comparison

The data model compare editor performs a three-way comparison if you compare three data models, or if you synchronize a data model that is under source code control.

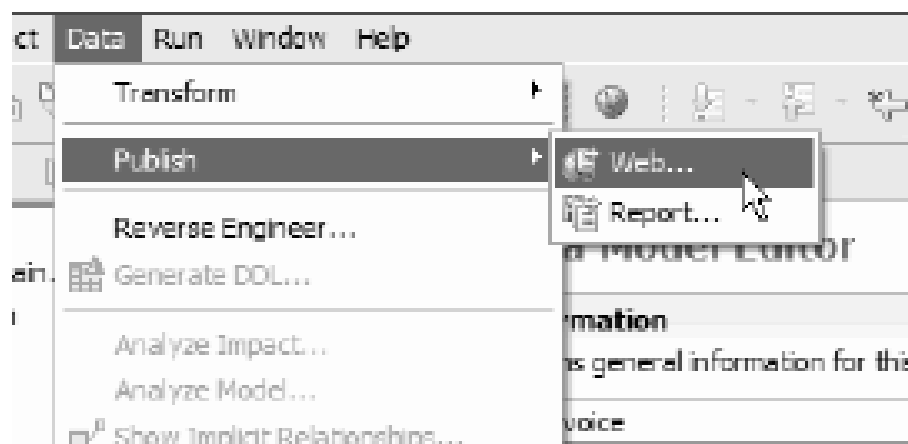


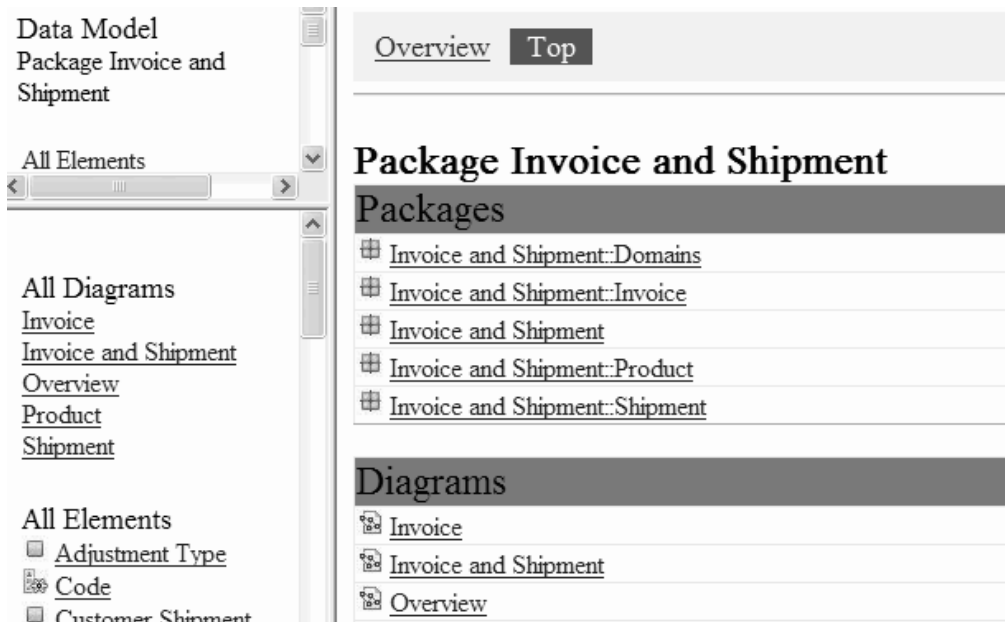
The differences between data objects are listed in the compare editor, so that you can browse and accept changes that were made by the contributors. You can merge changes directly in the compare editor and save the changes to the affected data models.

Web publishing and data model reporting

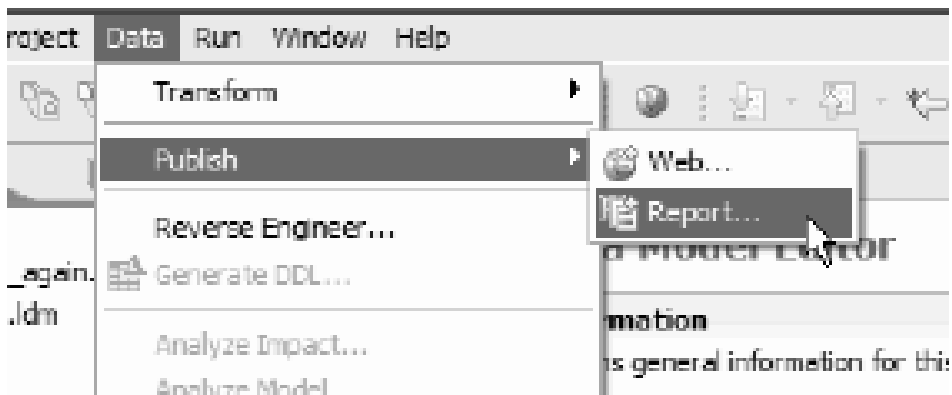
You can use the Web publishing and reporting features in Rational Data Architect to communicate model information to team members who might not have access to the workbench tooling.

You can publish data models to a hypertext-linked web of HTML pages that show information about the models. The published information mirrors the content of the models and packages, and you can view the published information by using a standard browser. The publishing process shows model elements, including diagrams, relationships, and attributes. You can use the published links to traverse the model the same way that you do in the workbench.





You can use the reporting feature to generate reports to help you analyze a data model or communicate model information to other team members. You can generate reports that use the open source Business Intelligence and Reporting Tools (BIRT) technology, or you can generate classic XSLT reports. You can customize the contents of a BIRT report, and you can choose between several formats including HTML, Microsoft® Word, Microsoft Excel, Microsoft PowerPoint, or PDF.



Logical Model Report For Invoice and Shipment

Model Information	
File Name	Invoice.ldm
File Location	C:\RDA_7005_workspace\Invoice\Invoice.ldm
File Size	161231
Last Modified	Sat Jun 21 14:04:31 PDT 2008

Identifying Relationship

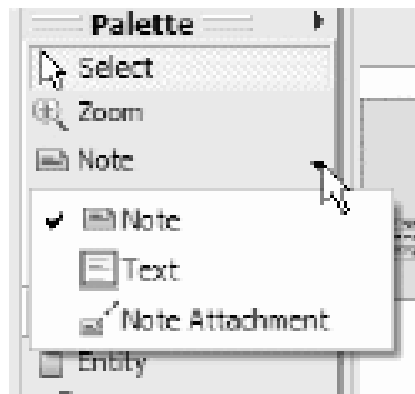
Relationship: Relationship1

Properties

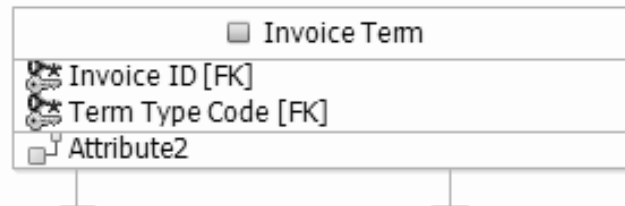
Is Identifying Relationship:	true
Parent Table:	Invoice
Parent Role Name:	Composed of
Parent Multiplicity:	ONE
Parent Referential Integrity: On Delete	CASCADE
Parent Referential Integrity: On Update	NONE

Data model diagram saving and printing

You can save data diagrams as an image file or print them, to facilitate communication between team members. You can also add annotations to data diagrams, such as notes and attachments, to communicate additional information about the diagram.



This note describes the entity below



Enterprise data modeling

You can use Rational Data Architect to model your database structures.

Logical and physical data modeling

Using data models helps to improve data quality in the following important ways:

- Providing a common semantic understanding of information
- Enhancing governance-defining data domains and enforcing enterprise standards
- Increasing productivity by automating physical database design and deployment tasks for multiple database servers
- Improving communication among team members
- Better managing complexity

Logical data models are not specific to a database. At a high level, they describe things about which an organization wants to collect data and the relationships among these things. Logical data models encapsulate business definitions, business objects, and industry best practices. They are organized hierarchically and contain objects such as packages, entities, attributes, and other relationship objects. They can also contain domain model objects, which describe the domain data types that an organization allows and their constraints. Use of domain models is described in more detail in the “Enforcement of enterprise standards” on page 15 section of this document.

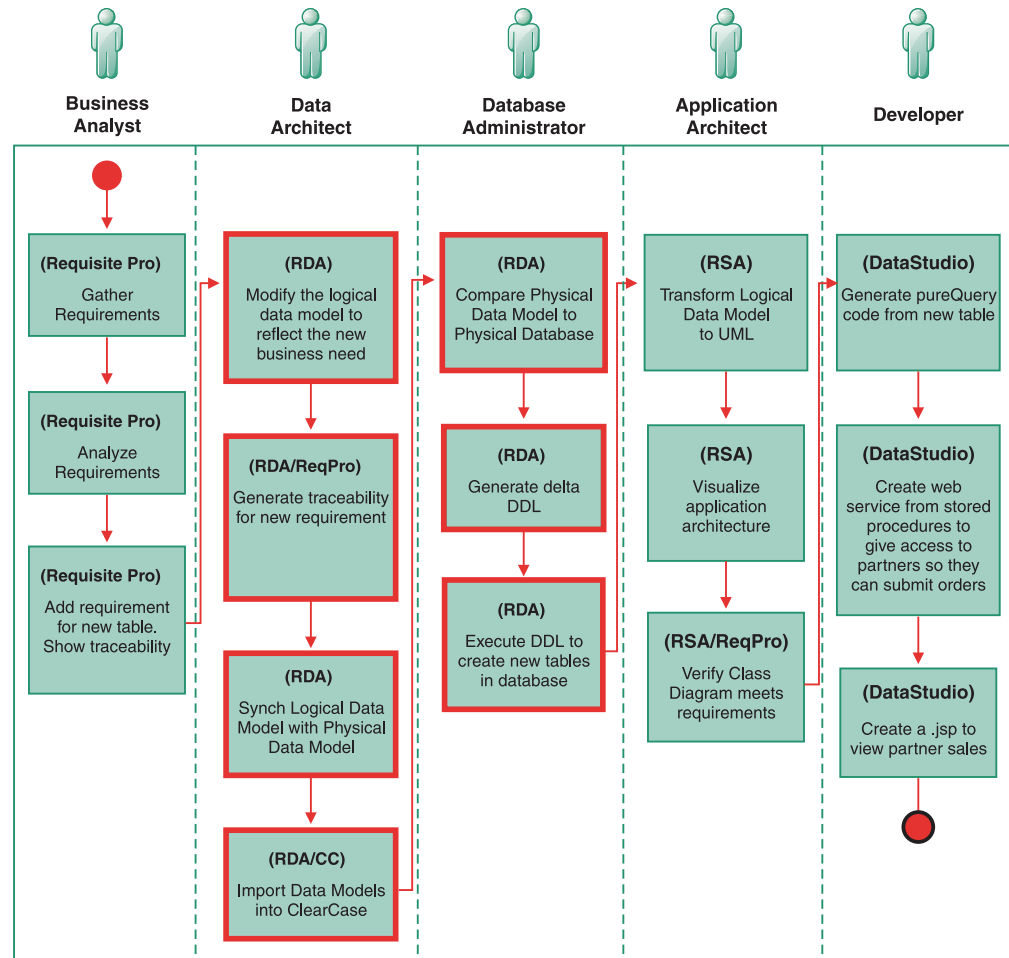
Logical data models are the semantic hub of enterprise architecture. Logical data models are sometimes overlooked in the software development life cycle, but they have become increasingly important in SOA for many reasons. A logical data model allows you to see an overview of data entities in an application or an enterprise without having to look at overwhelming implementation details. A logical data model is especially useful when you are dealing with increasingly complex and heterogeneous IT environments. A logical data model creates a common enterprise view of data, helps to reduce data redundancy, improves data quality, and speeds up integration and green-field projects. Other IT artifacts, such as service models, message models, object models, and physical data models, can be traced to logical data models semantically and often are transformed directly from logical data models.

Physical data models are database-specific models that represent relational data objects (for example, tables, columns, primary keys, and foreign keys) and their relationships. For some database targets, you can also use Rational Data Architect to manage storage objects in physical data models, such as table spaces and buffer pools.

Rational Data Architect supports multiple database servers, including:

- DB2® for Linux®, UNIX®, and Windows®
- DB2 for z/OS®
- DB2 for iSeries®
- Informix®
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

Rational Data Architect data modeling features fit into the application lifecycle as shown in the following picture:



Creating, visualizing, and modifying data models

Rational Data Architect provides tooling that you can use to easily generate, transform, view, and modify logical and physical data models. This tooling includes:

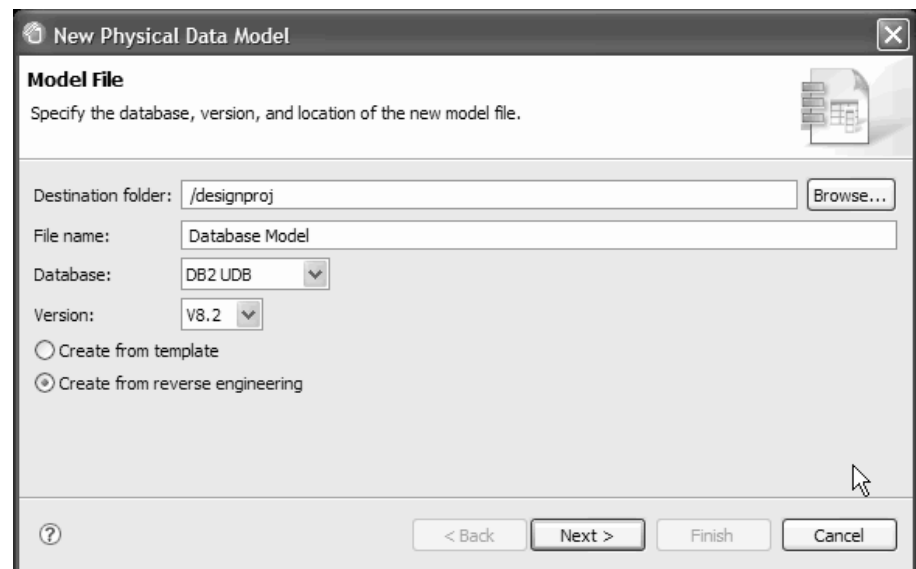
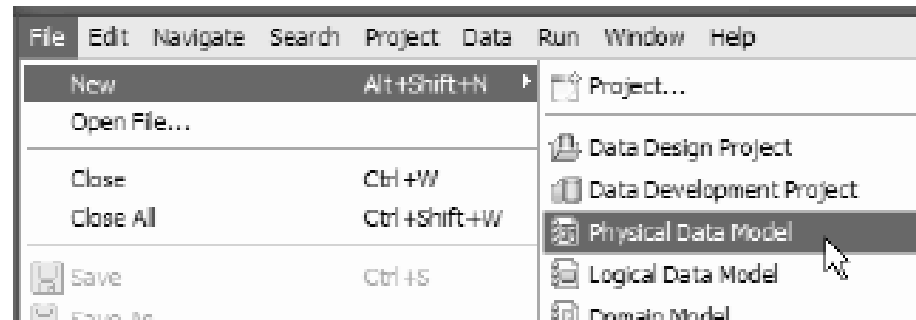
Drag and drop support

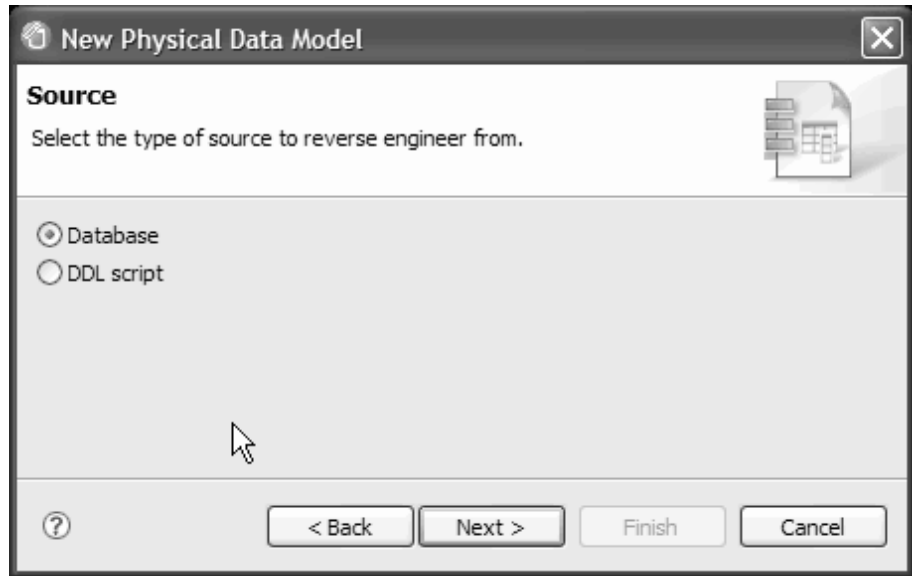
After you create a connection to a database in the Rational Data Architect Data Source Explorer, you can easily drag and drop a schema to a data design project to generate a physical data model that is created using default options.

You can likewise drag and drop physical data models to other data design projects, which creates a copy in the new project.

Reverse engineering with a wizard

If you want to modify default options for physical data model creation from an existing database, you can generate a physical data model by using a wizard. You can also reverse engineer a physical data model from a DDL script.





Template support

You can export a data model to a template file that can then be used to create new data models with a wizard. Templates are supported for logical and physical data models, as well as glossary models and domain models.

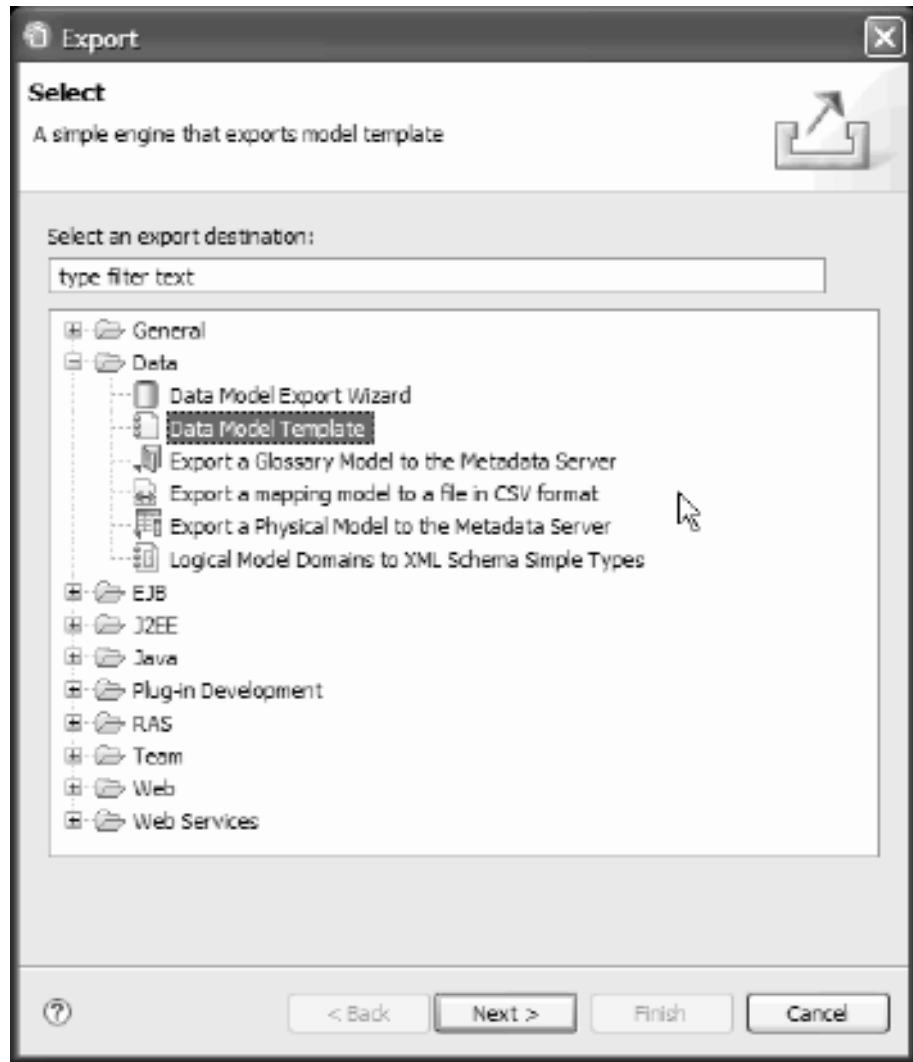
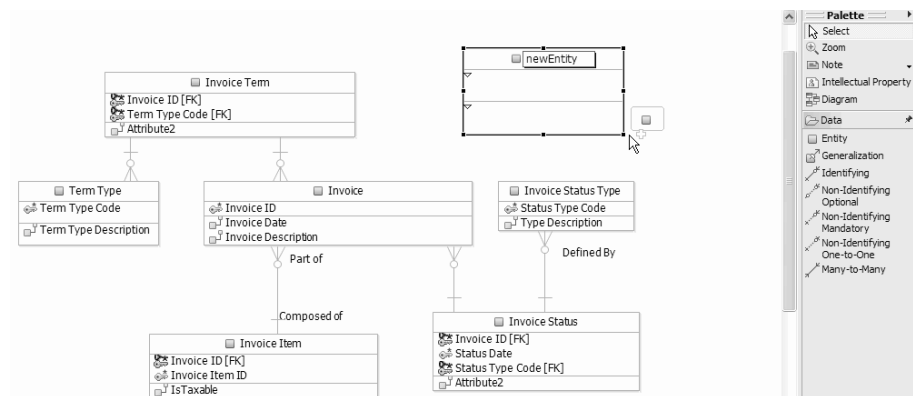


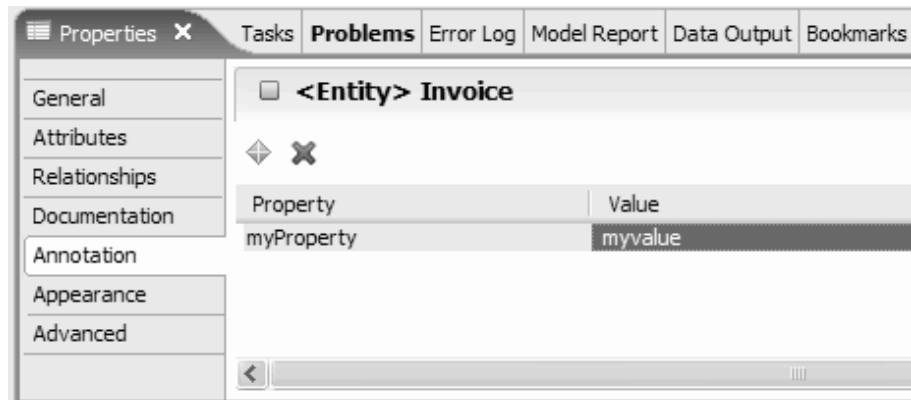
Diagram editor and Properties view

You can use information engineering (IE) diagrams to visualize and modify data models. With the diagram editor, you can easily drag and drop new data structures from a palette onto a model, or draw relationships between data objects.



You can also modify data model object properties in the tabbed Properties view as you view them in a diagram or the Data Project Explorer. You can

add additional annotations to the model by adding user-defined properties and values.



For more information about how Rational Data Architect can be used to address data in an SOA environment, read the following article: *The power of Rational Data Architect: An overview of Rational Data Architect and SOA* by Periasamy Girirajan at <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0708girirajan/>.

Enforcement of enterprise standards

You can use Rational Data Architect data standardization features to improve data quality and enhance understandability. These features are especially critical for large organizations with multiple team members.

Rational Data Architect has three main features that allow teams to enforce organizational standards:

Domain modeling

Domain models describe the domain types and their constraints that an organization allows. You can also annotate domain data types with masking attributes that you can use to generate test data from the data in a production database. Domain data types can be stored in a domain model or as part of a logical data model. You can associate a domain data model with a data design project so that the domain data types are available for either logical or physical data modeling. Using atomic domains instead of base data types ensures consistency among team members.

Glossary (naming) modeling

Data object naming standards promote a common understanding of data. You can use glossary models to enable sharing of data across organizational boundaries and reduce data redundancy through the consolidation of synonymous and overlapping data elements.

Data model analysis

At any time when you are building a data model, you can analyze the model to verify that it is compliant with the defined constraints. You can also analyze databases using this same functionality.

Enforcement of data type standards with domain models

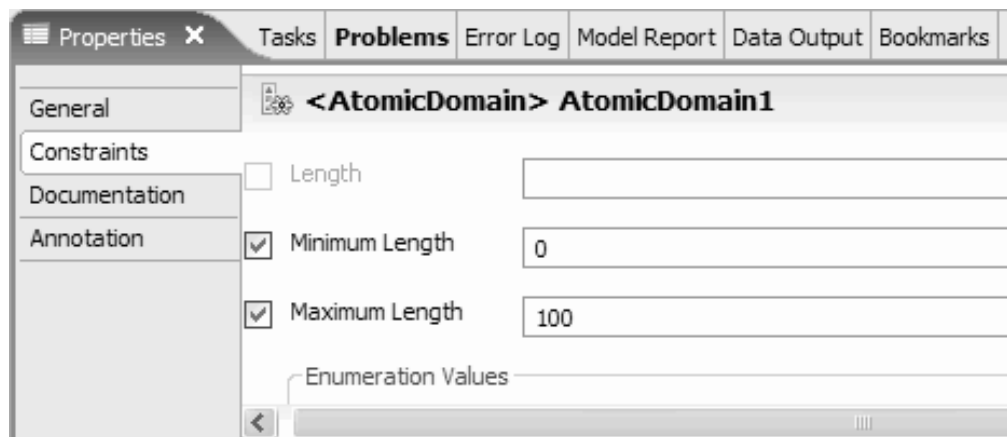
A domain model consists of a group of domain data type objects. A domain object represents an abstract data type that can be restricted by adding constraints.

Domain data types are based on base data types. For example, you can define domain data types for commonly used definitions such as social security number, sex, height, or marital status.

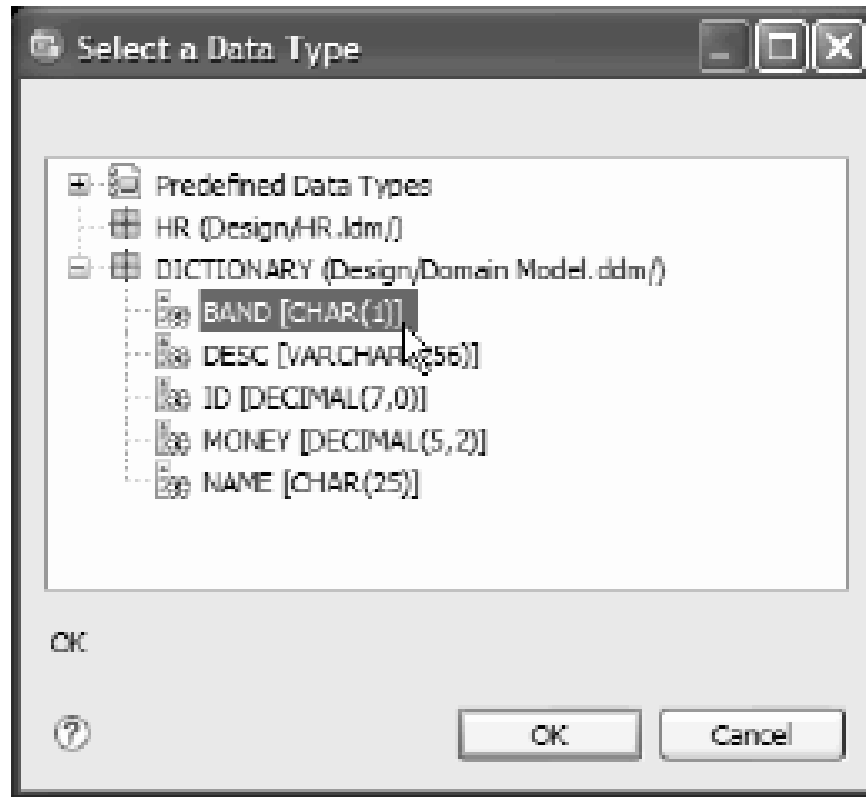
You can use Rational Data Architect to create domain models and atomic domains, and then share these models among team members who are working on logical or physical data models.

Rational Data Architect provides tooling that helps you create and manage domain models. You can create a domain model from a wizard, either from an existing template or from scratch.

Use the Data Project Explorer to add domain objects, then use the Properties view to modify them:



When you associate one or more domain models with a data design project, the domain data types in the model are available for selection when you are creating or modifying the data type of a physical or logical data model object.



Enforcement of naming standards with glossary models

When a naming standard is enforced, terms defined in a glossary must be used and term order must be followed when you are naming a data object. Rational Data Architect provides tools that help you define glossary and data object naming standards, to make it easy to create data object names that are compliant with these naming standards. You can also use a glossary model to validate existing data objects, either in a model or on a database, to ensure their compliance with naming standards.

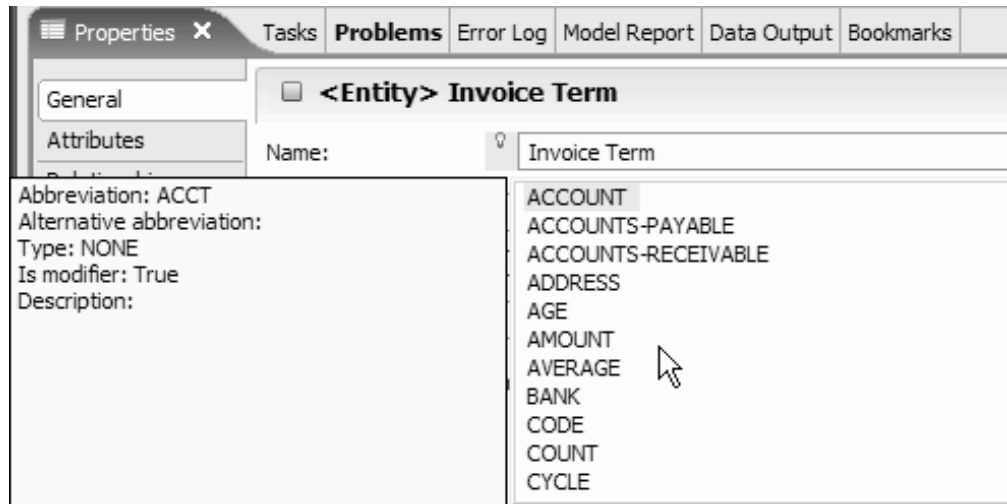
You can use a glossary model in the following ways:

- During the analysis of a data model, to determine whether the model adheres to naming standards.
- During data model object modification in the Properties view, to provide content assistance when specifying object names.
- During logical to physical data model transformation, to determine how to name physical data objects.
- During the creation of a mapping model, to allow the discover function to more easily relate source and target elements. Mapping models are described in more detail in “Data and information integration” on page 21.

You can use Rational Data Architect to generate glossary models, and then share these models among team members who are working on logical or physical data models.

Rational Data Architect helps you create and manage glossary models. You can create a glossary model from a wizard, either from an existing template or from scratch.

When you associate one or more glossary models with a data design project, the glossary model contents are available for selection when you are modifying a data object name in the Properties view.

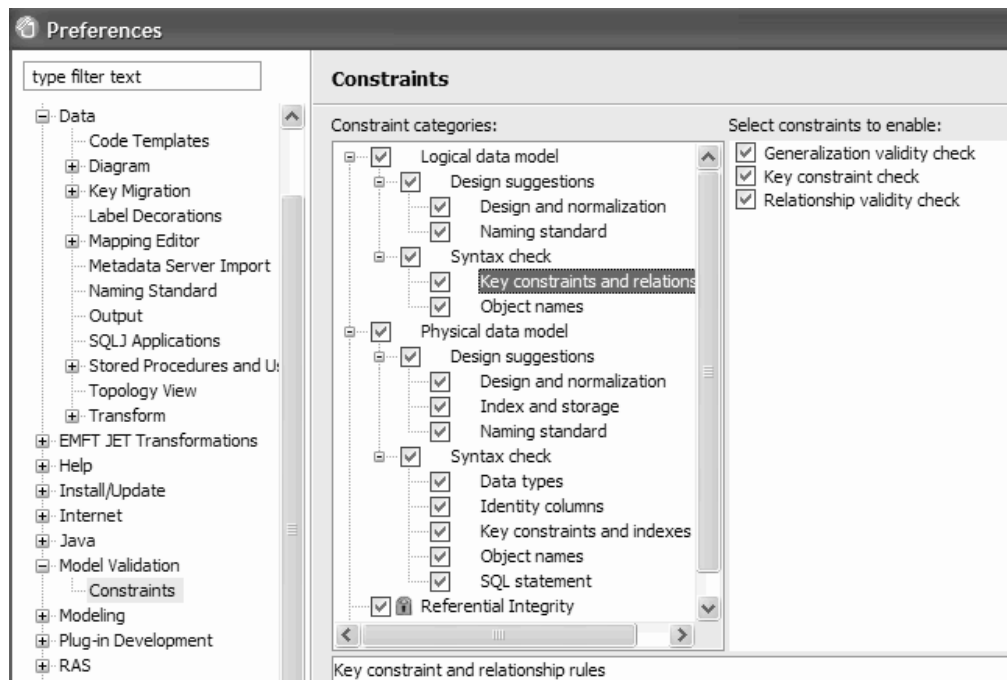


You can also use the glossary model during data model analysis, which is described below.

Enforcement of additional constraints with data model analysis

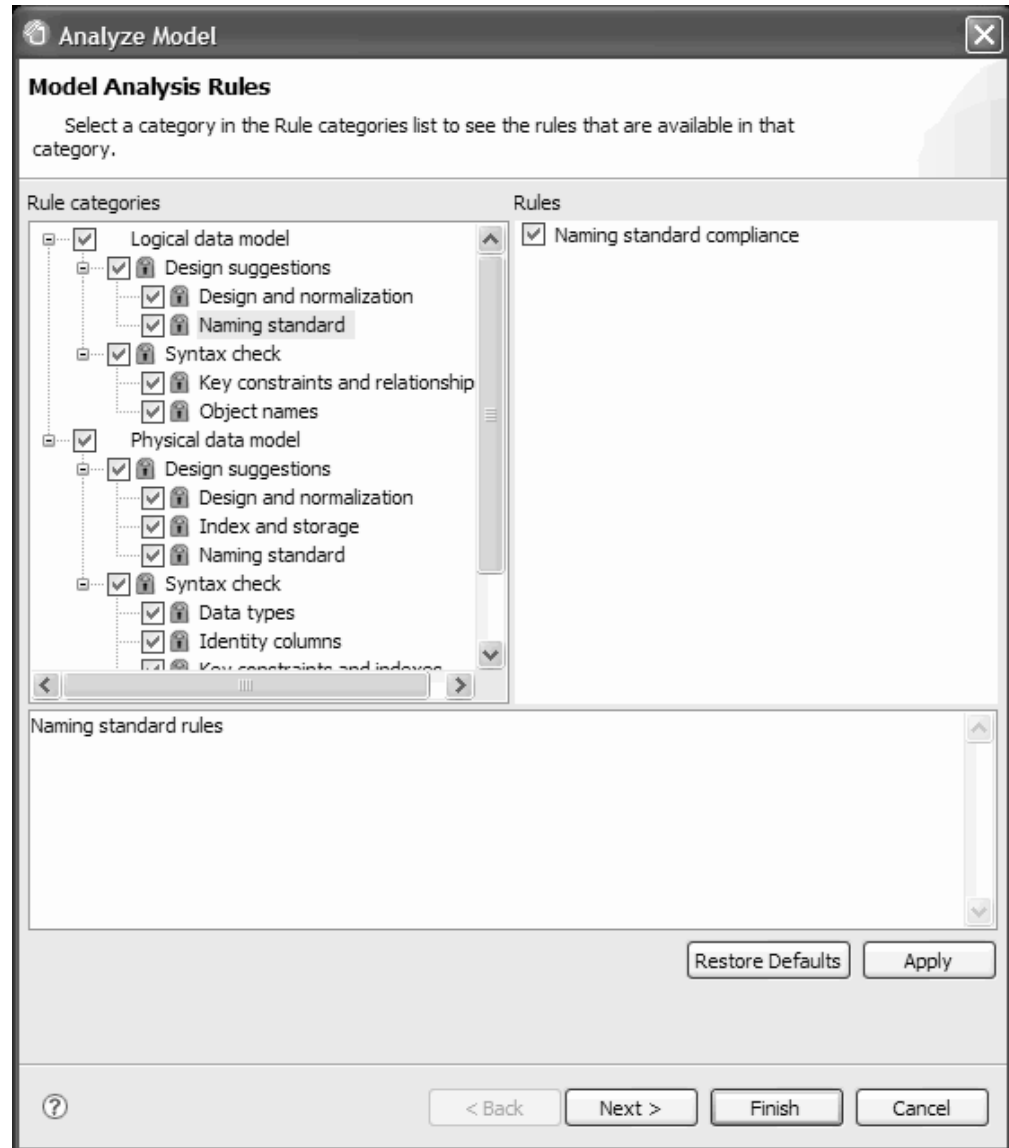
At any time when you are building a data model, you can analyze the model to verify that it is compliant with the defined constraints. You can also analyze existing databases. Before you analyze a data model, you can set preferences for data model analysis. Then you can use a wizard to specify additional model analysis settings and perform the analysis. Results are displayed in the Problems view, from which you can navigate to listed objects that are not in compliance.

You can set preferences for data model analysis in the Preferences window:



You can analyze a model that is stored in a data project, or you can analyze a database that is displayed in the Data Source Explorer.


In the wizard that opens, you can modify your global preferences for model validation:



Results are displayed in the Problems view. You can double-click an object that is displayed in the Problems view to navigate to the object so that you can fix the problem.

Properties Tasks Problems x Error Log Model Report Data Output Bookmarks		
1 error, 44 warnings, 1 info		
Description	Resource	Path
Warnings (44 items)		
⚠ Attribute Adjustment Description is not compliant with naming standards -- A Invoice.ldm	Invoice.ldm	Invoice
⚠ Attribute Adjustment Type Code is not compliant with naming standards -- A Invoice.ldm	Invoice.ldm	Invoice
⚠ Attribute Adjustment Type Code is not compliant with naming standards -- A Invoice.ldm	Invoice.ldm	Invoice
⚠ Attribute Amount is not compliant with naming standards -- A prime word is r Invoice.ldm	Invoice.ldm	Invoice
⚠ Attribute Attribute1 is not compliant with naming standards -- A prime word Invoice.ldm	Invoice.ldm	Invoice
⚠ Attribute Attribute2 is not compliant with naming standards -- A prime word Invoice.ldm	Invoice.ldm	Invoice

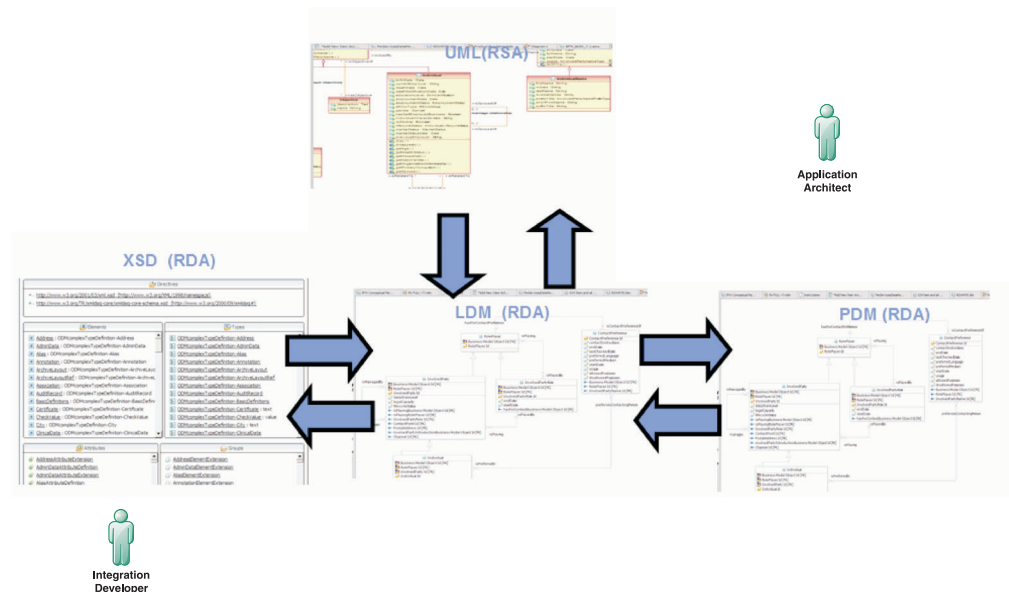
Related information

 IBM developerWorks article: Use Rational Data Architect to define and enforce data object naming standards

Transforming data models

Rational Data Architect data models can transform easily to and from other formats. This allows team members to communicate more effectively and translate business needs into database objects.

The "round trip" transformation process in Rational Data Architect ensures that every application component or service that is identified has an underlying data component. Rational Data Architect takes you seamlessly from high-level application requirements that are encapsulated in a UML model all the way to a physical database manifestation. Then you can generate XML views over the data for service oriented architectures, and easily communicate database changes back to the application or service architect. This type of communication is an important aspect of service oriented architecture (SOA).



You can easily transform data models to other formats by using wizards in Rational Data Architect. The following transformations are supported:

Logical to physical data model

Use this transformation to generate a database-specific physical data model

from a generic logical data model. Generate multiple physical data models that target different database servers, from the same logical data model. Update existing physical data models with changes that you make to a source logical data model. As part of the logical-to-physical data model transformation, you can specify transformation options to denormalize the model, either by transforming generalization subtypes and supertypes as separate tables, rolled up, or rolled down.

Physical to logical data model

Generate a generic logical data model from a database server-specific physical data model, which you can then propagate to other database servers.

Logical data model to UML model, and UML model to logical data model

Use these transformations to generate or validate database structures against application requirements in a UML model.

Logical data model to XML schema, and XML schema to logical data model

Use these transformations to generate XML schemas from logical data models, then modify and finalize the XML schema for use in defining the structure of XML documents or messages.

For more information about how Rational Data Architect can be used to address data in an SOA environment, read the following article: *The power of Rational Data Architect: An overview of Rational Data Architect and SOA* by Periasamy Girirajan at <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0708girirajan/>.

Data and information integration

Data integration across data sources in an enterprise of data sources is one of the most common requirements of any database shop. Users need on-demand integrated access to a variety of diverse data sources — from traditional data assets such as Excel spreadsheets and legacy data, to new and emerging technologies such as XML data stores. IBM InfoSphere Information Server provides a single interface that allows users and database administrators to virtually and transparently access a wide variety of these data sources. By using Rational Data Architect with InfoSphere Information Server, data architects and application developers can discover, model, visualize, relate, and develop data assets across distributed and diverse data sources.

You can use Rational Data Architect to perform the following data integration tasks:

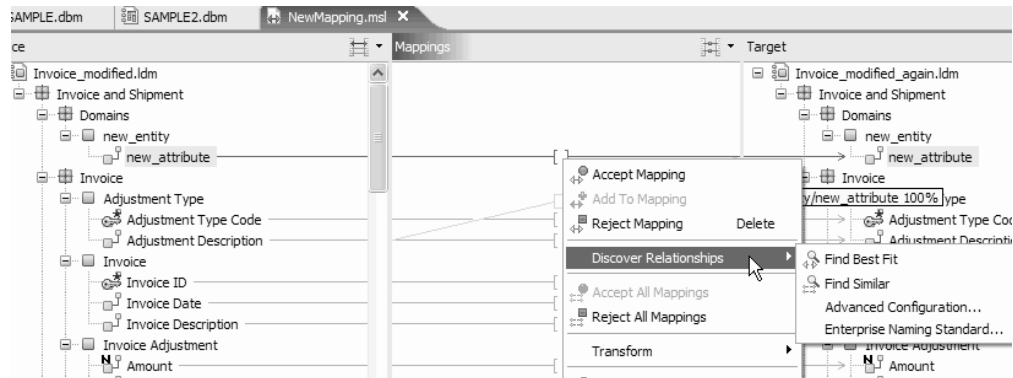
- Discover relationships between databases (source and target schemas or related tables)
- Map relationships between data models
- Build expressions that provide integrated views across the data objects
- Generate scripts from the mapping model that you can run to integrate the source and target

You can also access remote data sources that are defined to a federated database, use the Topology view to display a read-only graphical view of the peer relationships between objects in your enterprise, create nicknames for remote tables that are imported to the Data Project Explorer, and select tables that might be in multiple databases and transform the tables into one new XML schema model.

Information integration modeling

You can use the enterprise metadata tools that are provided with Rational Data Architect to find and map relationships among a variety of data sources, build scripts that represent those relationships, and then deploy scripts to federated, non-federated, local, or remote servers. Use InfoSphere Information Server to access diverse data and the processes that support the data. Then, use enterprise data modeling to interpret and exploit the metadata.

Rational Data Architect provides a mapping editor that you can use to visually define relationships between elements of one or more data sources and a single target in a data design project. The data source can be any physical data model or logical data model in the project. The target can be a physical data model, a logical data model, or an XML schema (XSD file).



You can use the mapping editor to implement "hub-and-spoke" information integration for:

- A data warehouse
- SOA-based information integration

The following information integration functions are provided:

Discover relationships

Provides different methods to show you how one or more source schemas might be related to a target schema. The relationships can be based on metadata, the naming information in a glossary model, or any available data samples.

Map relationships

Provides a visual representation of source and target data models and the relationships that are either discovered or manually specified. The mapping describes how you can combine and transform data that is represented by one or more source models into some form that is appropriate to a target model. The results are saved in a mapping model.

Build expressions

Provides you with a way to define criteria, such as functions, join conditions, filter conditions, and sort conditions, that you can add to the mapping model. You can then use the expressions in scripts that you deploy.

Generate scripts

Generates a script from the mapping model that you can deploy. You can use the script to transform and filter data from mapping-model-compliant sources to mapping-model-compliant targets.

Accessing remote data sources that are defined to a federated database

In the Data Source Explorer, you can access remote data sources that are defined to a federated database. Rational Data Architect supports connecting to a WebSphere Information Integrator federated database and accessing the remote data sources that are defined to the federated database.

Rational Data Architect supports JDBC access to the following data sources:

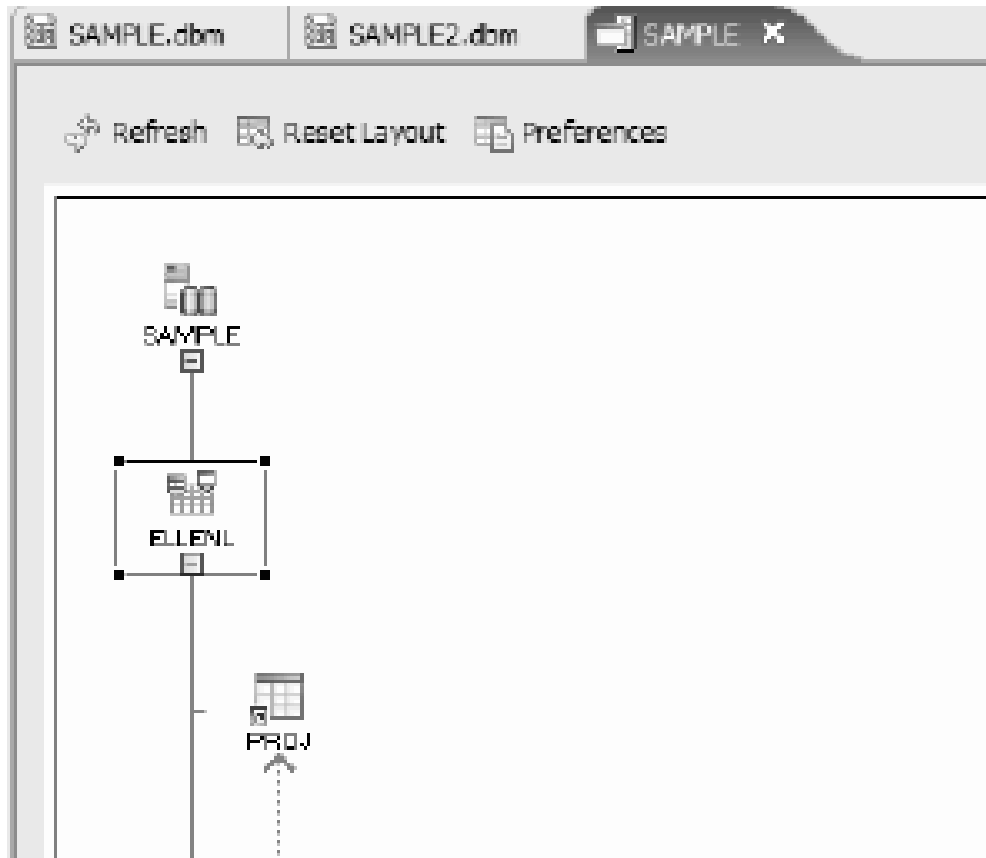
- DB2 family
- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

Rational Data Architect also supports ODBC data sources, including Excel and WebSphere II Classic Event Publisher (by using an ODBC connection)

Visualizing relationships between enterprise objects

The Topology view in Rational Data Architect displays a read-only graphical view of the peer relationships between objects in your enterprise. This view shows peer relationships between databases, schemas, and tables. For example, you can see all of the objects in an InfoSphere Information Server federated server and the federated databases that it connects to. You also can view the information for remote tables and determine whether nicknames are defined for those tables on the InfoSphere Information Server federated server.

You can print the contents of the Topology view to share this information with other team members.



Managing data and database objects

You can use Rational Data Architect to generate and run DDL scripts to create or modify data objects or data model objects.

Data modelers can also use the script generation and team collaboration features in Rational Data Architect to integrate their data modeling work with database administration tasks.

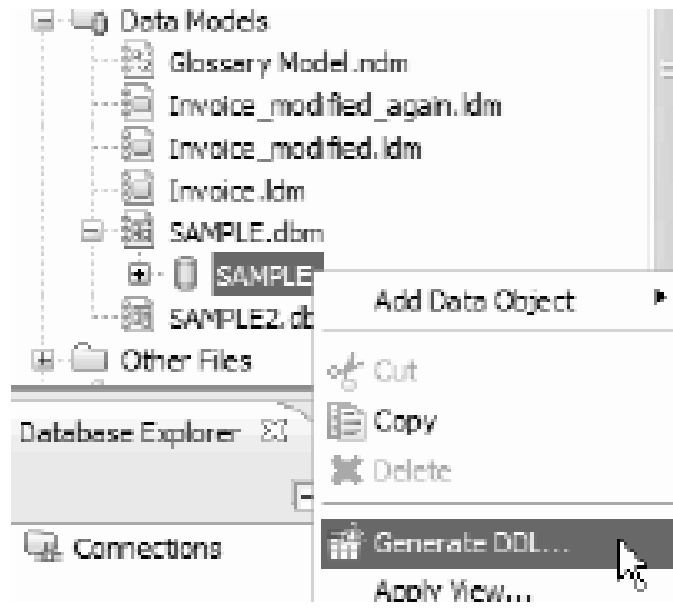
DDL script generation

You can generate and run a DDL script to:

- Keep a snapshot of the database structure
- Set up a test system where the database acts like the production system but contains no data
- Produce templates for new objects that you can create based on existing objects
- Make changes to database objects based on changes to data models

For example, generate the DDL for the Customer table, then edit the DDL to create the table Customer_New with the same schema.

You can generate DDL scripts from data objects that are displayed in the Data Source Explorer, or from data model objects in a data project.



After you generate the DDL script, you can customize it as needed by using the SQL editor.

 A screenshot of a SQL editor window. The window has three tabs: 'SAMPLE.dbm', 'SAMPLE2.dbm', and 'script.sql'. The 'script.sql' tab is active and shows the following SQL script:


```

DROP SCHEMA "ELLENL" RESTRICT;

DROP SCHEMA "SYSIBM" RESTRICT;

DROP BUFFERPOOL "IBMDEFAULTBP";

DROP DATABASE PARTITION GROUP "IBMDEFAULTGROUP";

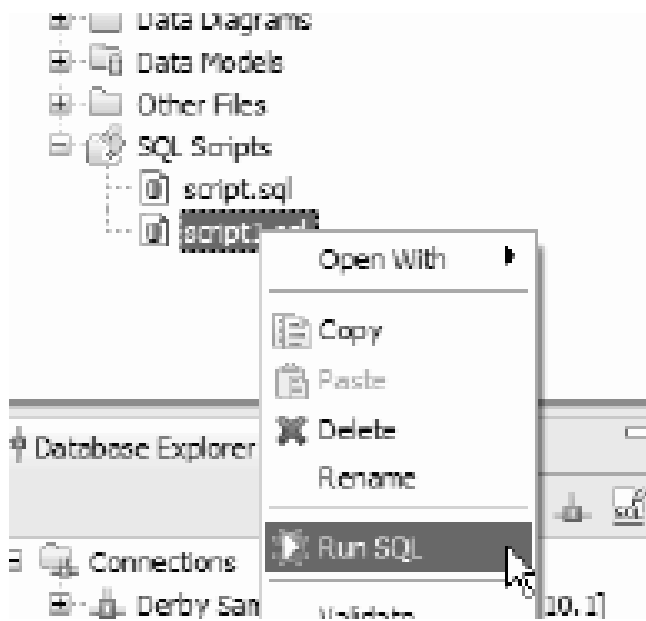
CREATE DATABASE PARTITION GROUP "IBMDEFAULTGROUP" ON DBPARTITIONNUMS (0);

CREATE BUFFERPOOL "IBMDEFAULTBP"
  ALL DBPARTITIONNUMS SIZE 250
  NUMBLOCKPAGES 0
  PAGESIZE 8 K;

CREATE SCHEMA "ELLENL";

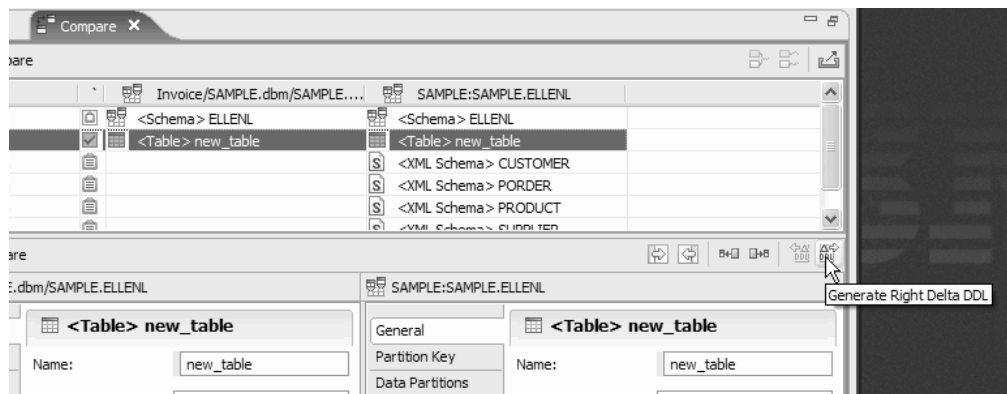
CREATE SCHEMA "SYSIBM".
  
```

When you generate a DDL script, you can choose to run the script directly after it is generated by selecting an option in the Generate DDL wizard, or you can simply save the script and run it later from a data project.



Delta DDL script generation in the compare editor

When you are merging changes in the compare editor, you can generate a delta DDL script that encompasses your changes. You can then modify or run the script to implement your changes on the database.



IBM Rational Data Architect user roles

Team members who perform different roles in the data design and development process can use Rational Data Architect together cohesively.

The following list describes data design and development team roles in an organization. Users in these roles can use Rational Data Architect to perform common tasks. In some cases, one person in an organization might perform a single role, and in other cases one person might perform multiple roles. In either case, Rational Data Architect provides tools that allow users in different roles to easily communicate and facilitate their task flow.

Data modeler

Data modelers partner with end users/client personnel to translate business needs into a data model that will be the basis for a solution that closely aligns with the business requirements. Data modelers are skilled in the repetitive process of refining data into precise data models that specify entities, attributes, and relationships between the data. Data modelers create the logical data model and assist the database administrator with translating the logical model into a workable physical model. Data modelers interact with users and client personnel via non-technical language to validate the data model by using the user's frame of reference. This frame of reference usually includes process models, use cases, and transactional requirements.

Data architect

Data architects design the elements that are required to structure the data aspects of solutions, including databases, file systems, and storage management. Data architects are responsible for metadata registries that are used to store and maintain metadata definitions. Data architects enforce standards and integrity within enterprise data models.

Data steward

Data stewards are responsible for maintaining a data element in a metadata registry. In some cases, this role is merged with that of a data architect. Data stewards are commonly needed when organizations are attempting to exchange data precisely and consistently between computer systems and reuse data-related resources.

Database administrator

Database administrators specify, place and modify data to support the development of applications solutions, and are responsible for the data that is used, updated, and produced by the application. Database administrators work with the data architect to transform the logical data model to the physical model and the resulting database structures. They help programmers and engineers to efficiently use the database. They troubleshoot production issues to meet service level agreements for performance and availability.

Enterprise architect

Enterprise architects work with stakeholders, both leadership and subject-matter experts, to build a comprehensive view of the organization's strategy, processes, information, and information technology assets. The enterprise architect links the business mission, strategy, and processes of an organization to its IT strategy, and documents everything with multiple architectural models or views that show how the current and future needs of an organization will be met.

Application architect

Application architects design, develop, and support application solutions to meet client requirements. Application architects might design, develop, or re-engineer highly complex application components, and integrate software packages, programs, and reusable objects that reside on multiple platforms.

Business analyst

Business analysts work as liaisons among stakeholders in order to elicit, analyze, communicate, and validate requirements for changes to business processes, policies, and information systems. Business analysts understand business problems and opportunities in the context of the requirements and recommend solutions that enable the organization to achieve its goals.

Integration architect

Information architects perform a sequence of activities in a disciplined manner in order to successfully develop an integrated information system.

Bibliography

Wikipedia contributors, "Data architect," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Data_architect (accessed June 14, 2008).

Wikipedia contributors, "Data steward," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Data_steward (accessed June 14, 2008).

Wikipedia contributors, "Database administrator," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Database_administrator (accessed June 14, 2008).

Wikipedia contributors, "Enterprise architect," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Enterprise_architect (accessed June 14, 2008).

Notices

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*Intellectual Property Dept. for Rational Software
IBM Corporation
3600 Steeles Avenue East
Markham, Ontario
Canada L3R 9Z7*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this documentation and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- developerWorks®
- IBM
- Passport Advantage®
- Rational
- WebSphere
- iSeries
- zSeries®

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names, may be trademarks or service marks of others.



Printed in USA