

**Modeling in Visio® Enterprise**

Copyright © 1997–1998 Visio Corporation. All rights reserved.

Information in these materials is furnished for informational use only, is subject to change without notice, and does not represent a commitment on the part of Visio Corporation. These materials, as well as the software described herein (“Software”), are furnished under license; there is no transfer of title. The Software is subject to the license agreement that accompanies or is included with the Software, which specifies the permitted and prohibited uses of the Software. Any unauthorized duplication or use of Visio Corporation Software, in whole or in part, in print, or in any other storage and retrieval system is prohibited. No part of these materials may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means (electronic, mechanical, recording, or otherwise) for any purpose other than the purchaser’s personal use without the express written permission of Visio Corporation. Visio Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in these materials. Use these materials at your own risk.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, VISIO CORPORATION AND ITS SUPPLIERS DISCLAIM ANY AND ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, AND THOSE ARISING OUT OF USAGE OF TRADE OR COURSE OF DEALING, CONCERNING THESE MATERIALS. THESE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL VISIO CORPORATION OR ITS SUPPLIERS (OR THEIR RESPECTIVE AGENTS, DIRECTORS, EMPLOYEES, OR REPRESENTATIVES) BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, ECONOMIC, PUNITIVE OR SIMILAR DAMAGES, OR DAMAGES FOR LOSS OF BUSINESS PROFITS, LOSS OF GOODWILL, BUSINESS INTERRUPTION, COMPUTER FAILURE OR MALFUNCTION, LOSS OF BUSINESS INFORMATION, OR ANY AND ALL OTHER COMMERCIAL OR PECUNIARY DAMAGES OR LOSSES) ARISING OUT OF THE PURCHASE OR USE OF THESE MATERIALS, HOWEVER CAUSED AND ON ANY LEGAL THEORY OF LIABILITY (WHETHER IN TORT, CONTRACT, OR OTHERWISE), EVEN IF VISIO CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Unless otherwise noted, all names of companies, products, street addresses, data, characters, and persons contained herein are part of a completely fictitious scenario or scenarios, are designed solely to document the use of a Visio Corporation product, and are in no way intended to represent any real individual, company, product, or event.

#### Third-Party Technology Credits:

ImageStream Graphics Filters copyright © 1998 by INSO Corporation. All rights reserved.

International CorrectSpell spelling correction system copyright © 1995 by Lernout & Hauspie Speech Products N.V. All rights reserved.

Portions copyright © 1991–1996 Arthur D. Applegate

Portions of this program copyright © 1982–1995 Pervasive Software Inc. All Rights Reserved.

Certain LZW graphics capability licensed from Unisys Corporation under U.S. Patent No. 4,558,302 and foreign counterparts.

Some of the clip art used in this product is derived from images copyrighted ©1988-1995 3G Graphics, Inc. from its IMAGES WITH IMPACT!® FOR WINDOWS™ Vol. 1. These images are used here under a non-exclusive licensing agreement between Visio Corporation and 3G Graphics, Inc., 114 Second Ave. S., Suite 104, Edmonds, WA 98020, USA, (206) 774-3518 or (800) 456-0234.

The maps incorporated into this product are derived from images copyrighted by Cartesia Software. These images are used here under a non-exclusive licensing agreement between Visio Corporation and Cartesia Software, P.O. Box 757, Lambertville, NJ 08530, USA, (609) 397-1611.

Visio Corporation Trademarks: Visio, VisioModeler, Visio Solutions Library, SmartShapes, ShapeSheet, SmartConnectors, SmartLayers, InfoModeler, ActiveQuery, the Visio corporate logo, and the Four Shapes logo are either registered trademarks or trademarks of Visio Corporation in the United States and/or other countries. AutoDiscovery and Guided Updating are either registered trademarks or trademarks of Kaspia Systems, Inc., a wholly owned subsidiary of Visio Corporation, in the United States and/or other countries.

Third-Party Trademarks: All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

US Government Restricted Rights: These materials are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. The contractor/manufacturer is Visio Corporation, 520 Pike Street, Suite 1800, Seattle, WA 98101-4001, USA.

Visio Corporation  
520 Pike Street, Suite 1800  
Seattle, Washington 98101-4001  
USA

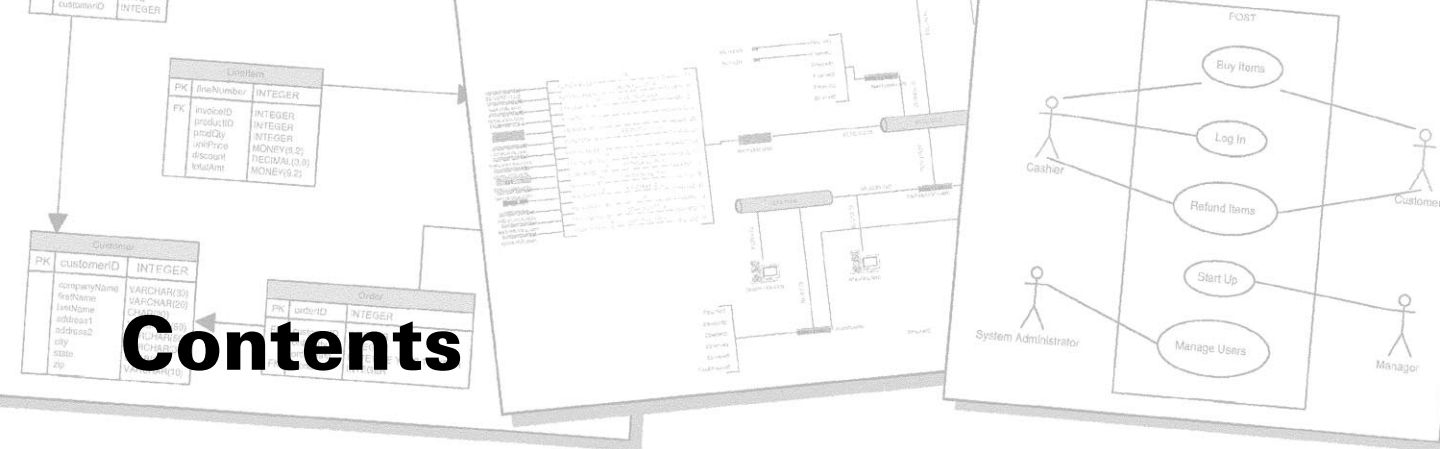
Visio International Limited  
Fitzwilton House, Wilton Place  
Dublin 2  
Ireland

Corporate telephone: (206) 521-4500  
Corporate fax: (206) 521-4501

International telephone: +353 1 6612036  
International fax: +353 1 6612047

Printed in USA.

Part No. 13704-0998



<b>Preface .....</b>	<b>1</b>
Who this manual is for .....	2
About this manual .....	2
Other manuals in the set .....	3
Getting more information .....	4
 <b>Chapter 1 Introduction .....</b>	<b>5</b>
Modeling in Visio Enterprise .....	6
Working with Visio Enterprise solutions .....	7
Sharing models with other Visio users .....	8
Solutions for diagramming and documenting networks .....	9
Installing AutoDiscovery And Layout .....	9
Installing Visio Network Equipment .....	11
Installing the network diagram solutions .....	11
Installing the Novell Directory Services solution .....	12
Solutions for modeling software .....	13
Installing UML .....	13
Installing the software diagram solutions .....	14
Solutions for modeling databases .....	16
Installing Database Modeling .....	16
Installing the database diagram solutions .....	18
Installing VisioModeler .....	18
Installing ActiveQuery .....	19
Next steps .....	20

<b>Chapter 2</b>	<b>Discovering and diagramming a network .....</b>	<b>23</b>
	Discovering your network .....	24
	How long does discovery take? .....	25
	Using the Basic Discovery Wizard.....	25
	Using the Advanced Discovery Wizard.....	26
	Creating a network diagram .....	27
	Creating a one-page diagram.....	27
	Creating a hyperlinked diagram .....	28
	Adding detail to a diagram using lists .....	30
	Device lists .....	30
	Device counter lists .....	31
	DLCI lists .....	31
	Interface lists .....	32
	Updating your diagram.....	33
	Using VNE or other shapes.....	35
	Changing text colors for network devices.....	40
	Managing the AutoDiscovery database.....	41
	Using multiple AutoDiscovery databases .....	43
<b>Chapter 3</b>	<b>Using network shapes.....</b>	<b>45</b>
	Working with network shapes.....	46
	Using the Locate Network Equipment utility.....	47
	Using the Visio Network Equipment Directory .....	48
	Creating a physical network diagram.....	49
	Working with labels .....	51
	Connecting 2-D shapes .....	52
	Creating reports.....	53
	Modifying a network shape and using the VNE Property Duplicator .....	54

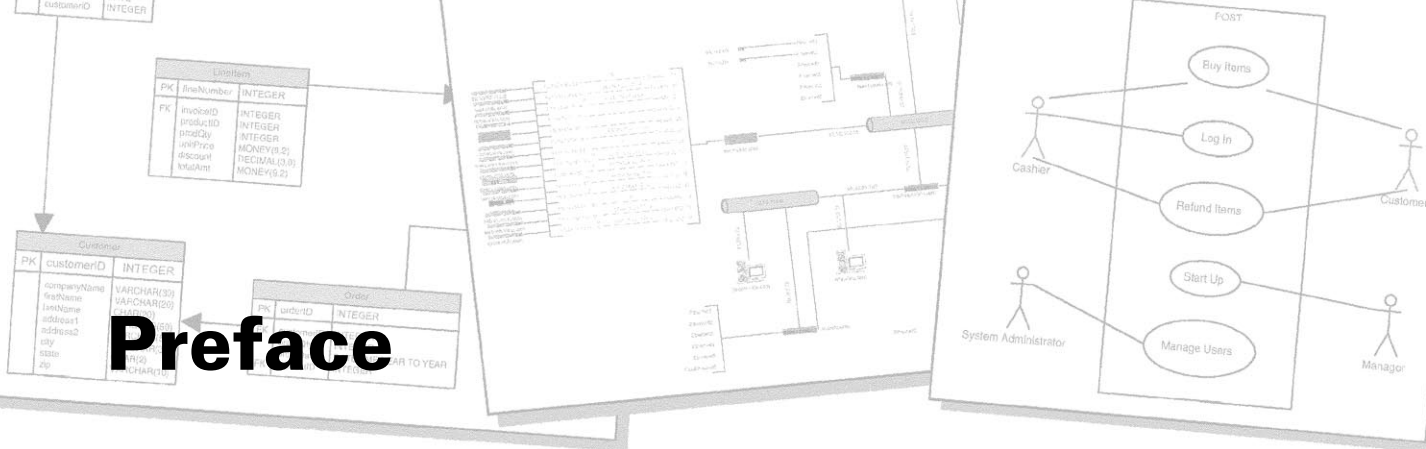
<b>Chapter 4</b>	<b>Modeling software components .....</b>	<b>59</b>
	Modeling software components in Visio Enterprise .....	60
	Software modeling solutions.....	61
	The Unified Modeling Language in Visio Enterprise .....	63
	One UML drawing file represents a system.....	65
	One system includes many models.....	66
	View one model in different ways.....	67
	Comparing the tree structure and the diagrams .....	68
	Using packages .....	69
	Adding property values to UML elements .....	70
	Conforming to the UML syntax .....	71
	Getting help.....	72
<b>Chapter 5</b>	<b>Creating software models with the UML.....</b>	<b>73</b>
	How to model a system using the UML solution .....	74
	Modeling a system using the UML .....	75
	Modeling a new system .....	76
	Adding a new model to an existing system .....	78
	Starting a new diagram within a model.....	79
	Adding UML elements to a model.....	79
	Associating property values with UML elements.....	82
	Diagnosing and fixing semantic errors.....	83
	The UML diagrams.....	84
	Phase 1: use case model .....	84
	Phase 2: domain model .....	86
	Phase 3: design model.....	91
	Phase 4: implementation model .....	97
	Storing and retrieving models: working with Microsoft Repository .....	99
	Importing data from Microsoft Repository .....	100
	Synchronizing versions of a model.....	100
	Exporting data to Microsoft Repository.....	101
	Reverse engineering source code .....	102
	Microsoft Visual C++.....	102
	Microsoft Visual Basic.....	104

<b>Chapter 6</b>	<b>Understanding the database modeling process.....</b>	<b>109</b>
	About the database modeling process in Visio Enterprise.....	110
	Working in a Visio Enterprise diagram .....	112
	Using the Database menu .....	113
	Working with Microsoft Repository .....	114
	Creating a database model.....	116
	Refining a database model .....	120
	Customizing a model's on-screen appearance .....	121
	Using right-click shortcut menus.....	122
	Using the Code window, Tables & Views window, and Types window.....	122
	Validating a model .....	123
	Connecting to a database.....	124
	Generating a database .....	124
	Updating a model or a database.....	125
	Creating a report .....	127
	Next steps.....	128
<b>Chapter 7</b>	<b>Creating database models.....</b>	<b>129</b>
	Creating a new database model .....	130
	Representing relational and object-relational databases.....	130
	Setting display options for a database model.....	132
	Creating a database model from scratch.....	133
	Reverse engineering an existing database .....	135
	Connecting to a data source .....	136
	Reverse engineering code.....	138
	Extracting a schema using the Reverse Engineer Wizard.....	139
	Extracting a database schema from Microsoft Repository.....	140
	Importing a data model .....	141
	Customizing your database model diagram .....	142
	Next steps.....	144

<b>Chapter 8</b>	<b>Refining database models.....</b>	<b>145</b>
	Working with tables .....	146
	Using categories .....	150
	Working with columns.....	152
	Working with data types .....	154
	Setting data types .....	155
	Using collection types and composite data types in a database model.....	159
	Working with views .....	161
	Working with relationships .....	164
	Setting referential integrity and cardinality.....	166
	Creating and editing indexes.....	168
	Setting extended attributes.....	171
	Validating a database model .....	173
	Next steps .....	174
 <b>Chapter 9</b>	 <b>Reverse engineering database code.....</b>	 <b>175</b>
	Viewing and editing code.....	176
	Working in the Code Editor .....	178
	Adding code to a model .....	180
	Creating stored procedures, functions, and other global code .....	181
	Defining view code .....	182
	Creating trigger code .....	184
	Creating check clauses .....	184
	Saving and synchronizing code.....	186
	Transferring code to a new target platform .....	188
	Next steps .....	189
 <b>Chapter 10</b>	 <b>Connecting to a DBMS.....</b>	 <b>191</b>
	About using database drivers .....	192
	Enhanced database drivers in Visio Enterprise .....	192
	Before you connect .....	193
	Configuring and associating a Visio Enterprise driver .....	194
	Creating and setting up an ODBC data source.....	196
	Connecting to Microsoft Repository .....	198
	Next steps .....	199

<b>Chapter 11</b>	<b>Generating a database schema .....</b>	<b>201</b>
	Preparing to generate a database or a DDL script .....	202
	Before you generate .....	202
	Choosing a generate option for a DBMS .....	203
	Generating and editing a DDL script .....	204
	Generating a new database .....	207
	Next steps.....	210
<b>Chapter 12</b>	<b>Updating models and databases .....</b>	<b>211</b>
	Keeping a model and database in sync.....	212
	Updating a model from a database .....	214
	Updating a database from the model .....	215
	Connecting to a data source .....	217
	Running the Update Model Wizard.....	218
	Running the Update Database Wizard .....	220
	Creating a script that updates a database .....	220
	Updating a database schema directly .....	223
	Next steps.....	225
<b>Chapter 13</b>	<b>Creating reports .....</b>	<b>227</b>
	Generating reports .....	228
	Formatting report titles and pages .....	230
	Previewing a report .....	233
	Exporting a report .....	234
	Summarizing file and database statistics in a report .....	234
	Listing database tables in a report .....	236
	Listing data types in a report .....	238
	Next steps.....	239
	<b>Glossary.....</b>	<b>241</b>
	<b>Index .....</b>	<b>249</b>





# Preface

Whether you model software components, design databases, implement network changes, or manage corporate information technology—or do some of each—the Visio® Enterprise modeling solutions can help you efficiently manage and maintain your systems and processes. With Visio Enterprise, you can design, model, and manage enterprise-level systems, such as enterprise network backbones, wide and local area networks (WANs and LANs), network premises infrastructures, software applications, and database management systems (DBMSs).

*Modeling in Visio Enterprise* is part of an integrated print and online documentation set designed to present the information you need to model your networks, software components, and databases.

## Topics in this chapter

- Who this manual is for .....2
- About this manual .....2
- Other manuals in the set .....3
- Getting more information .....4

# Who this manual is for

*Modeling in Visio Enterprise* is written for the corporate information technologist, consultant, or system integrator who is responsible for modeling and managing enterprise-level systems. This manual assumes that you use models and diagrams to communicate information as part of your job, and that you have used other Microsoft Windows–based applications and are familiar with Windows terminology and techniques.

This manual focuses on the specific solutions, shapes, and templates in Visio Enterprise that you use to design and model networks, software components, and DBMSs. Visio Enterprise also includes a wide variety of customizable tools that you can use to communicate across the enterprise, from process flowcharts to floor plans to block diagrams. For details about these and other drawing types, as well as techniques for working in any diagram, refer to *Using Visio Enterprise*.

# About this manual

*Modeling in Visio Enterprise* describes how to use the enterprise modeling solutions in Visio Enterprise. The manual is organized into parts, and each part describes the solutions and tools available for modeling one enterprise area: network architecture, application development, or DBMSs. This organization enables a user working in one area to focus only on the information specific to that area, without needing to refer to information about the other solutions available in Visio Enterprise unless desired.

The parts in this manual provide the following information:

**Part 1 “Designing a Network with Visio Enterprise”** Explains how to use the AutoDiscovery™ technology in the Visio Enterprise AutoDiscovery And Layout solution together with the Visio Network Equipment (VNE) add-on to locate network devices and lay out network diagrams.

**Part 2 “Modeling Software with Visio Enterprise”** Describes how to create integrated visual system models using the Unified Modeling Language (UML) and reverse engineer information models to generate UML static structure models.

**Part 3 “Modeling a Database with Visio Enterprise”** Explains how to work in the Database Model solution to design, document, reverse engineer, generate, and update a data model and its corresponding schema.

## Other manuals in the set

Visio Enterprise also includes the *Using Visio Enterprise* and *Developing Visio Solutions* printed manuals. You can find the following information in each:

### *Using Visio Enterprise*

- Instructions for installing Visio Enterprise. (For details, see Chapter 1, “Getting started.”)
- Information about contacting product technical support. (For details, see “Preface.”)
- Conceptual discussions, step-by-step procedures, and helpful tips about how to work efficiently in any type of enterprise diagram.
- Complete information for creating your own shapes, stencils, and templates.

### *Developing Visio Solutions*

- Everything you need to know about Visio Enterprise as a development platform, including sample code, tips, and techniques.
- Details about the ShapeSheet® spreadsheet and other tools for customizing Visio shapes.
- Details about controlling shapes, pages, menus, and other objects using a programming language, such as Microsoft Visual Basic for Applications (VBA), that supports Automation.

**NOTE** When you install Visio Enterprise, you have the option to install user manuals in PDF format for the VisioModeler™ and ActiveQuery® software programs, as well as other documentation. For details about these and other installation options, refer to *Using Visio Enterprise*.

## Getting more information

Much of the assistance you need as you use Visio Enterprise is specific to the shapes and templates you work with on the screen. To give you immediate and targeted information as you work, use these online sources:

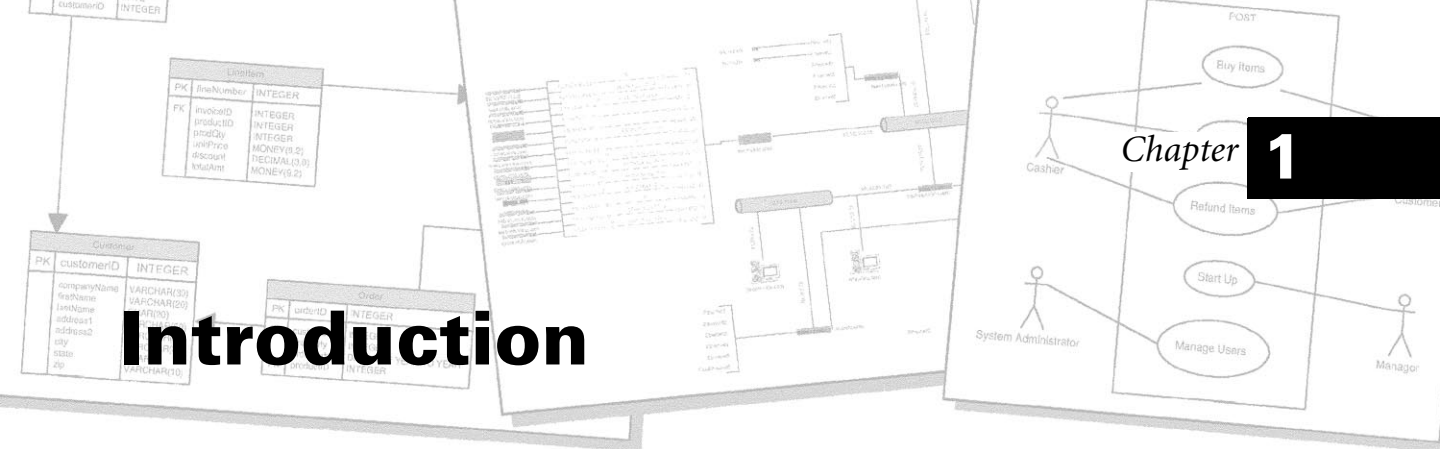
**Tips** To find out what a specific tool or button on one of the toolbars does, float the pointer over it for a moment. A tip appears on the screen.

**Shape-specific help** For details about the specific features of any Visio Enterprise shape, right-click the shape, then choose Shape Help from the shortcut menu. (You can print the topic by right-clicking the shape help window, then choosing Print.)

**Template-specific help** To discover the best sequence in which to work in a specific template, such as the UML Model Diagram or Database Model template, choose Help > Template Help, double-click Visio Templates, then choose the drawing type about which you want information.

**Visio Enterprise help** Help is available on the screen when you press F1, choose a command from the Help menu, or click Help in a dialog box. The online help also presents information that does not appear in this manual, including a command, tool, and programming language reference. For assistance with technical issues, you can also search on “Technical Support” in online help to locate technical support phone numbers for your area.

**Information, support, and service on the Web** To reach Visio Corporation using the World Wide Web, choose Help > Visio On The Web > Visio Home Page. Or, for information on technical issues, support, or to locate the Visio Knowledge Base, use the Visio address: <http://www.visio.com/support/>. The Visio Knowledge Base includes articles that answer frequently asked questions (FAQs) and offer the latest tips on using Visio products.



# Introduction

Visio® Enterprise is for information technology professionals who develop, manage, and maintain a complex enterprise information architecture. If you are designing new information systems, or maintaining and modifying existing ones, Visio Enterprise helps you model your business requirements by providing a unique combination of solutions for

- **Network design and management** Locate and diagram the enterprise network and all its components, from router to hub to workstation.
- **Software application modeling** Model the software development process in a variety of methods, including the Unified Modeling Language (UML).
- **Database design and modeling** Design, generate, validate, and re-engineer databases.

The network, software, and database modeling solutions in Visio Enterprise build on the strength of the Visio drawing engine to go beyond diagramming. These solutions offer a new level of visualization, design, and management of information architecture that retains all the flexibility associated with Visio products. This chapter introduces these specialized solutions and directs you to additional resources for more information.

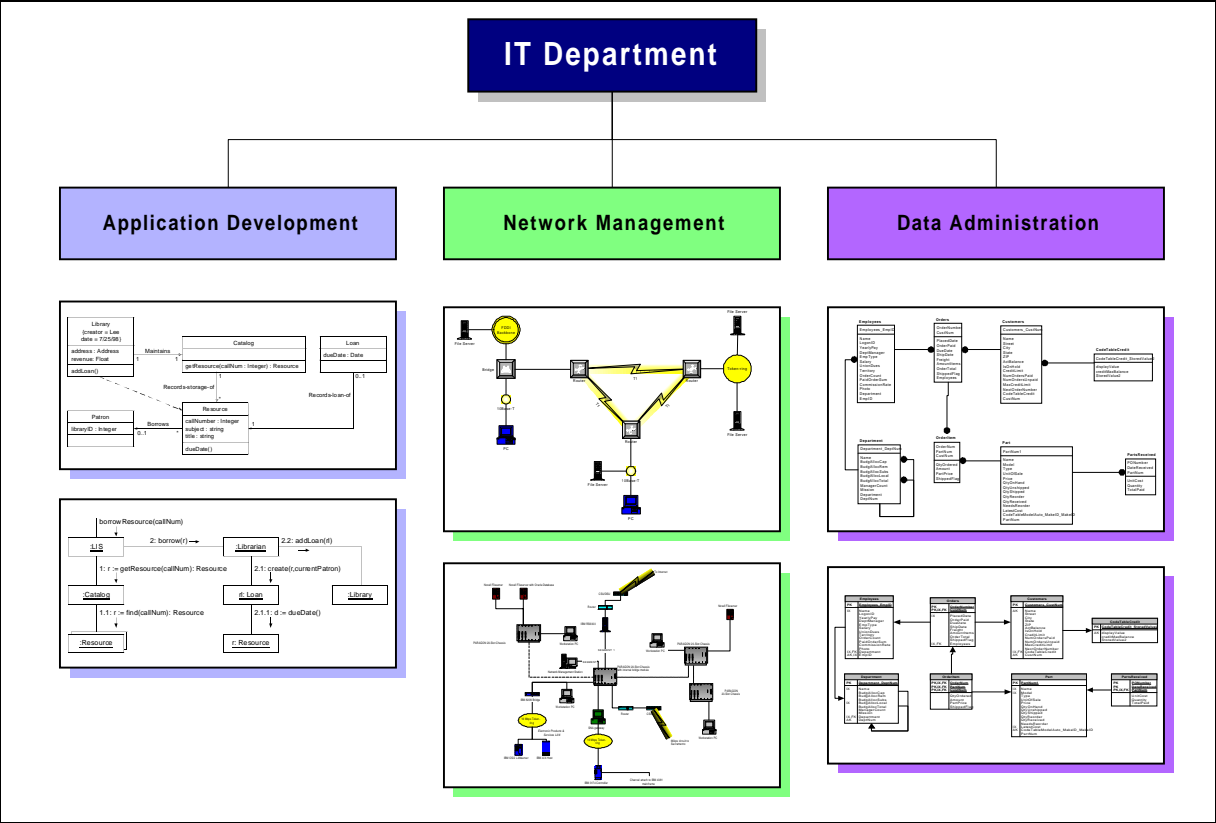
## Topics in this chapter

- Modeling in Visio Enterprise ..... 6
- Solutions for diagramming and documenting networks ..... 9
- Solutions for modeling software..... 13
- Solutions for modeling databases ..... 16
- Next steps..... 20

# Modeling in Visio Enterprise

With Visio Enterprise, you receive a suite of in-depth design and modeling tools for the three primary information technology functions: network management, software engineering, and database design. You can design and document the enterprise network backbone and wide and local area networks (WANs and LANs). You can create conceptual models for application development and reverse engineer code. You can reverse and forward engineer databases from the leading database vendors. By importing from and exporting to Microsoft Repository, you can extend and reuse your models, share them with colleagues, and port them to complementary enterprise tools.

**Figure 1-1** The tools in Visio Enterprise support all information technology areas.



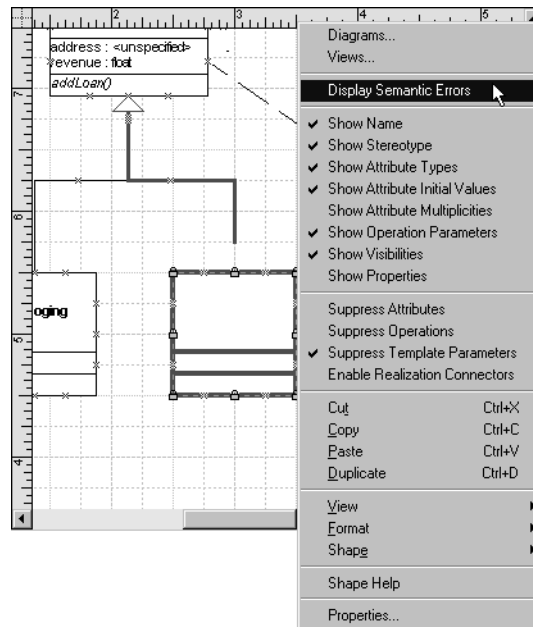
## Working with Visio Enterprise solutions

To design network, software, and database information architectures, you need tools that can represent conceptual, logical, and physical views of your enterprise. The solutions in Visio Enterprise provide these tools in a combination of add-on programs, stand-alone applications, and templates (.vst files). For example, Visio Enterprise includes the Visio Network Equipment (VNE) add-on with its thousands of equipment shapes, the stand-alone VisioModeler™ data modeling program, and network, software, database, and other templates that you can use to create models and diagrams.

As with other Visio solutions, you start a network diagram, software model, or database model by opening a template containing all the tools you need to work with: a drawing page with predefined page settings, stencils of solution-specific shapes, and preset text and graphic styles. These templates are designed with specialized menu commands, wizards, and shape behavior that are available only when you start a model or diagram with the appropriate template. For example, to use the Advanced Discovery Wizard, you must open the AutoDiscovery And Layout template. To define a foreign key relationship in a database, you must work with the Relationship and Entity shapes in the Database Model template. To check for errors in a UML model, you must start your model with the UML Model Diagram template.

**Figure 1-2**

Shapes in a model are more than pictures of enterprise components. Models in Visio Enterprise provide built-in behaviors, such as error checking, that let you accurately represent system functions.



The UML Model Diagram template is an example of a template that adds commands to Visio Enterprise that help you create better models.

The network, software, and database shapes are designed to work in a drawing started with a particular template, but you can use the shapes in a drawing started from a different template. However, you may lose some of the shape's special behavior. By starting with the appropriate template, you are assured of having access to all the built-in commands and behavior designed for the model or diagram. For details about how shapes, stencils, and templates work together, refer to *Using Visio Enterprise*.

## Sharing models with other Visio users

Visio Enterprise is designed to fill a unique niche in a corporate enterprise that can also include other Visio products. Models and diagrams created in Visio Enterprise can be opened and edited by users of Visio® Standard, Visio® Professional, and Visio® Technical.

In addition to the specialized tools for modeling and designing enterprise infrastructures, Visio Enterprise includes features common to all Visio products that make it the program of choice for visualizing and communicating ideas. Those features include

- Easy-to-use drag and drop drawing.
- SmartShapes® technology—shapes programmed to behave the way you expect.
- Compatibility and consistency with Microsoft Windows 95 and later, Office 97 and later, and Windows NT 4.0.
- A quick and easy process for publishing drawings as Web pages.
- A built-in development tool, Microsoft Visual Basic for Applications (VBA), that you can use to customize the Visio interface and program your own solutions.

For details about these and other tools common to all Visio products, refer to *Using Visio Enterprise* or see online help.



# Solutions for diagramming and documenting networks

The network solutions in Visio Enterprise provide a bridge between your networks and the variety of data sources, MIBs (management information bases), SNMP (Simple Network Management Protocol), and other applications typically used to depict physical and logical networks. These solutions allow you to

- Use AutoDiscovery™ technology to locate network components and create up-to-date network diagrams.
- Represent your network with exact-replica equipment shapes.
- Directly link with Novell™ Directory Services.

In the Visio Enterprise Setup program, you have the option of installing the following network solutions:

- AutoDiscovery And Layout, which installs the AutoDiscovery And Layout template
- VNE Program Files and Manufacturers, which installs Visio Network Equipment (VNE) add-on and selected manufacturer-specific shapes
- Network Diagram, which installs a set of network diagramming templates
- Novell Directory Services, which installs the NDS Solution template

The following sections describe the network components that you can install in the Visio Enterprise Setup program. For details about installation, refer to Chapter 1, “Getting started” in *Using Visio Enterprise*.

## Installing AutoDiscovery And Layout

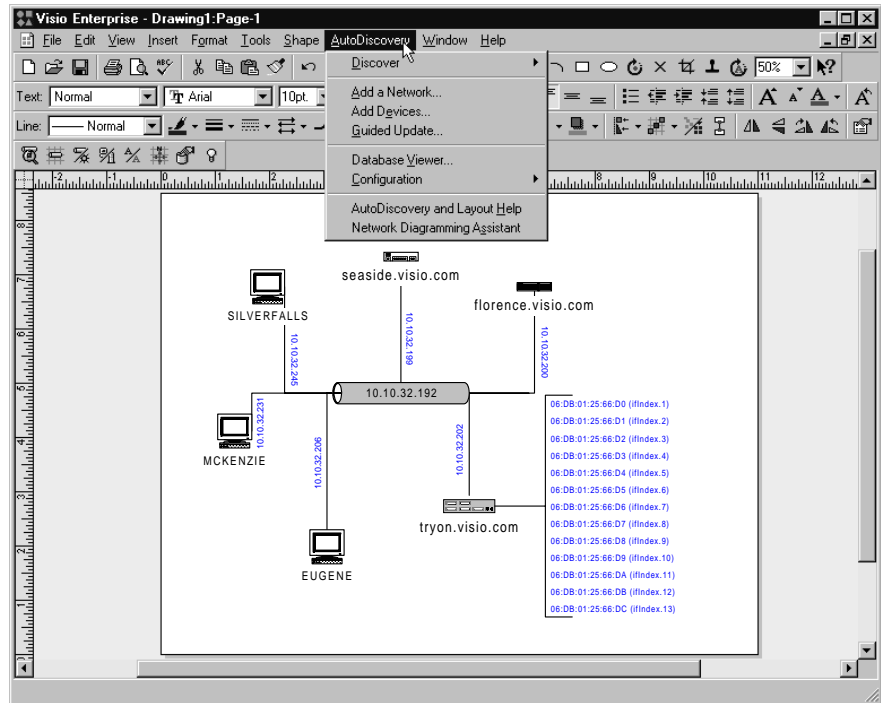
If you choose the AutoDiscovery And Layout option in the Setup program, you install the AutoDiscovery And Layout template. With this template, you can automatically locate all the devices in your LAN or WAN, then diagram them. Powerful wizards use the unique AutoDiscovery technology, which discovers network devices for you, then creates a diagram of the found equipment. Without network interruption, this SNMP technology starts with a single router to build a comprehensive database of your network, including both device and connectivity information, at the LAN or enterprise level.

Using the information captured during the discovery process, Visio Enterprise guides you through the automatic generation of logical network diagrams complete with connectivity and specific device interface information. Found devices are matched with vendor-specific shapes from the Visio Network Equipment library if you installed this option or with generic network shapes.

To keep up with changes to your network, the AutoDiscovery And Layout solution guides you through selective updating that reflects all changes that have occurred since the last discovery. You can quickly select only the portions of your network you need to update.

**Figure 1-3**

With the AutoDiscovery And Layout template, you can automatically generate a network diagram and interface list.



The AutoDiscovery And Layout solution is installed in the \\Visio\\Solutions\\Network Diagram folder.

**To start a drawing with the AutoDiscovery And Layout template:**

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click Network Diagram, click AutoDiscovery And Layout, then click OK.

For details about working with AutoDiscovery technology, see [Chapter 2, “Discovering and diagramming a network.”](#)

## Installing Visio Network Equipment

In the Visio Enterprise Setup program, VNE and the VNE shapes are Network installation options. Depending on the installation options you choose, you can install network equipment shapes from the top 10 manufacturers (Typical installation), or choose from the complete set of more than 14,000 manufacturer-specific shapes (Custom installation).

To find a particular network shape as you work, you can use the Locate Network Equipment utility, an installable option that opens automatically when you start a drawing with a VNE template. The Locate Network Equipment utility replaces the Visio stencil window with a shape database manager that allows you to quickly locate a specific network shape among the thousands available. This utility works only with the VNE shapes.

VNE templates are installed in the \Visio\Solutions\VNE folder.

### To start a drawing with a VNE template:

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click VNE; click Closet.vst, LAN.vst, or WAN.vst; then click OK.

For details about working with the equipment shapes, see [Chapter 3, “Using network shapes.”](#)

## Installing the network diagram solutions

The network diagram solutions provide a complete set of basic network diagram and office layout tools, as well as specialized templates for diagramming directory structures. In the Visio Enterprise Setup program, when you choose the Network Diagram option, you install the following templates:

- **Active Directory** Includes shapes compliant with Microsoft Active Directory icons for you to use in designing accurate and efficient directory structures.
- **Basic Network** Provides basic topology and equipment shapes for quick diagrams.
- **Logical Network Diagram and Logical Network Diagram 2** Allows you to create logical diagrams that describe the structure and scope of networks.
- **Office Layout For Networks** Includes office furniture and wall shapes for physical network diagrams.

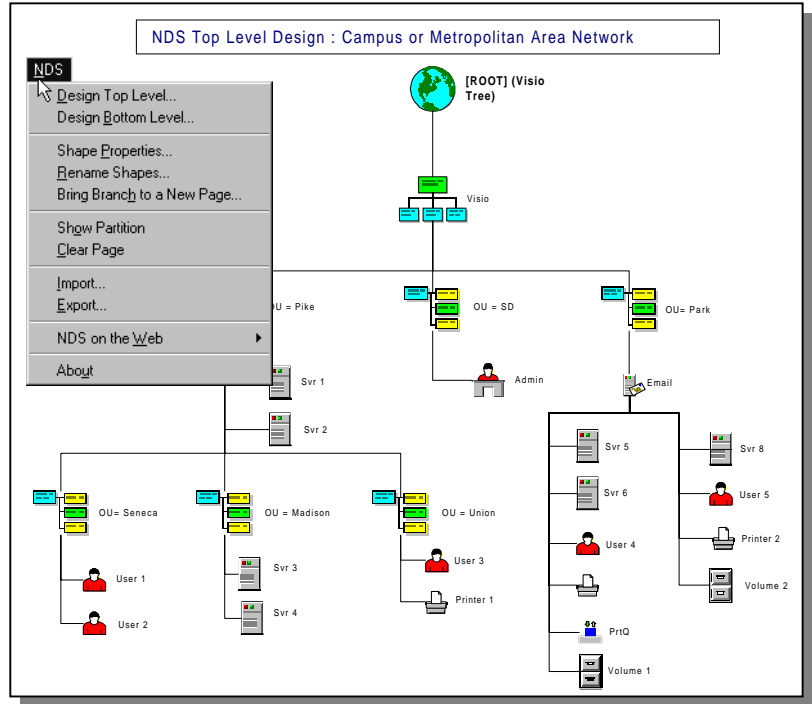
The network diagram solutions are installed in the \Visio\Solutions\Network Diagram folder. For details about these templates, choose Help > Template Help, double-click Visio Templates, double-click Network Diagram, select the one you want, then click Display.

## Installing the Novell Directory Services solution

The Novell Directory Services (NDS) Solution directly links to Novell directory structures. With more than 40 Container and Leaf objects, this template allows you to diagram existing NDS trees automatically and export new or updated tree views directly to NDS.

**Figure 1-4**

A tree diagram created with the Novell Directory Services template, which adds the NDS menu to the Visio Enterprise menu bar



The NDS Solution template is installed in the \Visio\Solutions\Network Diagram folder.

### To start a drawing with the NDS Solution template:

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click Network Diagram, click NDS Solution, then click OK.

For details about using NDS, choose Help > Template Help, double-click Visio Templates, double-click Network Diagram, then double-click NDS Solution.

# Solutions for modeling software

In the Visio Enterprise Setup program, you have the option of installing the following software solutions:

- UML, which installs the UML Model Diagram template
- Software Diagram, which installs a set of software templates representing a wide range of notations and methodologies templates (.vst files)

Visio Enterprise also installs Microsoft Repository 2.0.

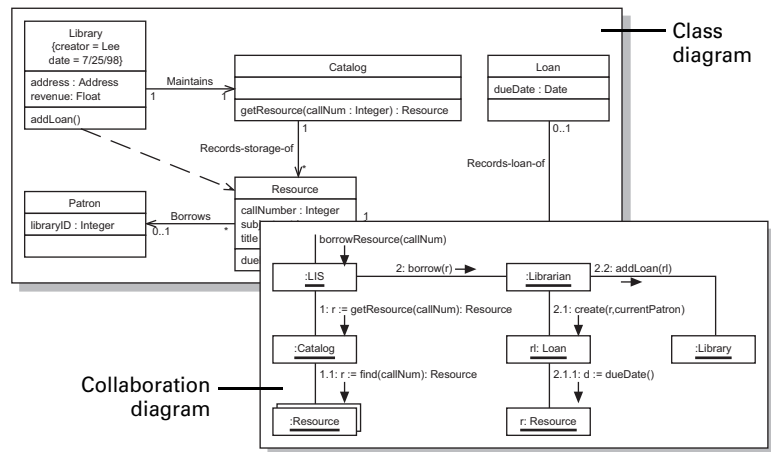
The following sections describe the software components that you can install in the Visio Enterprise Setup program. For details about installing Visio Enterprise, refer to Chapter 1, “Getting started” in *Using Visio Enterprise*.

## Installing UML

In the Visio Enterprise Setup program, UML is one of the Software installation options. If you choose UML, Setup installs the UML Model Diagram solution, which provides unparalleled support for the UML version 1.2. With this solution, you can

- Model in the entire UML 1.2 notation (documented in the *OMG Unified Modeling Language Specification*, included in PDF format with Visio Enterprise).
- Reverse engineer Visual C++ and Visual Basic code into UML models.
- Automatically check your model for syntax and semantic errors to ensure that your model meets UML 1.2 standards.
- Browse your model with the UML Navigator, which shows the entire UML model while you zoom in on the drawing page to edit its components.
- Store and retrieve UML models, diagrams, classes, and objects with Microsoft Repository 2.0.
- Import and export Microsoft Repository class definitions to view them graphically.

**Figure 1-5**  
With the UML Model Diagram template, you can create use case, domain, and implementation models as well as the design models shown here.



The UML Model Diagram solution is installed in the \Visio\Solutions\UML folder.

#### To start a model with the UML Model Diagram solution:

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click UML, click UML Model Diagram, then click OK.

For details about working with the UML tools in Visio Enterprise, see [Chapter 5](#), “Creating software models with the UML.”

### Installing the software diagram solutions

You can install software diagram solutions that provide a complete set of tools for designing, analyzing, and diagramming software projects. In the Visio Enterprise Setup program, when you choose the Software Diagram option, you install the following templates:

- **Booch OOD** Supports the creation of object, class, timing, state, module, and process diagrams using Booch object-oriented design notation.
- **Chen ERD** Supports the creation of entity relationship and data structure diagrams using Chen ERD notation.
- **COM And OLE** Supports the creation of COM (Component Object Model) objects and interfaces and system diagrams.

- **Express-G** Supports the creation of entity-level and schema-level diagrams using Express-G notation, the graphical component of the Express formal information requirements specification language.
- **Fusion** Supports the creation of diagram types included in Fusion methodology.
- **Gane-Sarson DFD** Supports the creation of data flow diagrams using Gane-Sarson notation.
- **Jackson** Supports the creation of data and program structures using the Jackson software design method.
- **Jacobson Use Cases** Supports the creation of use case, state transition, interaction, and other diagrams included in the Jacobson object-oriented software engineering method.
- **Martin ERD** Supports the creation of object-oriented analysis and design diagrams using Martin ERD notation.
- **Nassi-Schneiderman** Supports the representation of sequence, selection, case, and repetition in program structure charts.
- **Program Structure** Supports the creation of structural diagrams of programs and memory objects.
- **ROOM** Supports the creation of Real-Time Object-Oriented Modeling (ROOM) diagrams.
- **Rumbaugh OMT** Supports the creation of object, dynamic, and functional diagrams using the Rumbaugh Object Modeling Technique.
- **Shlaer-Mellor OOA** Supports the creation of class diagrams, class structure charts, dependency diagrams, and inheritance diagrams using Shlaer-Mellor notation.
- **SSADM** Supports the creation of logical data structures, data flow diagrams, entity life histories, and other system analysis and design diagrams.
- **System Structure** Supports the creation of diagrams that represent Windows and Macintosh operating systems and user interfaces.
- **Windows User Interface** Includes shapes for developing, documenting, and prototyping Office 97 and Windows 95 user interfaces.
- **Yourdon And Coad** Supports the creation of object state, data flow, and other diagrams included in the Yourdon And Coad object-oriented analysis and design notation.

The software diagram solutions are installed in the \Visio\Solutions\Software Diagram folder.

**To start a drawing with a software diagram template:**

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click Software Diagram, click the template you want, then click OK.

For details about working with the software modeling tools provided with Visio Enterprise, see [Chapter 4, “Modeling software components.”](#) For details about working with a particular template, choose Help > Template Help, double-click Visio Templates, double-click Software Diagram, select the one you want, then click Display.

## Solutions for modeling databases

Visio Enterprise provides database design and re-engineering tools for database systems from the leading vendors, including IBM, Informix, Microsoft, Oracle, and Sybase.

In the Visio Enterprise Setup program, you have the option of installing the following database solutions:

- Database Modeling, which installs the Database Model template
- Database Diagrams, which installs the Bachman and Object Role Modeling templates
- ActiveQuery, which installs the full ActiveQuery® program
- VisioModeler, which installs the full VisioModeler program

The following sections describe the database components that you can install in the Visio Enterprise Setup program. For details about installing Visio Enterprise, refer to Chapter 1, “Getting started” in *Using Visio Enterprise*.

### Installing Database Modeling

Visio Enterprise provides complete relational modeling and re-engineering capabilities through multiple notations with the Database Model solution. If you choose the Database Modeling option in the Setup program, you install this solution. Its tools allow you to

- Create entity relationship and object-relational data models, and switch between modeling notations on the fly.
- Reverse engineer databases and synchronize models with databases.



- Edit and define stored procedures, triggers, views, and check clauses.
- Perform detailed semantic and physical error checking for models.
- Create customizable reports.
- Import and export Microsoft Data Warehousing Framework (Microsoft Repository) models.
- Import models from VisioModeler, PLATINUM ERwin, and Visio Professional 5.0.

The Database Model solution is installed in the \Visio\Solutions\Database folder.

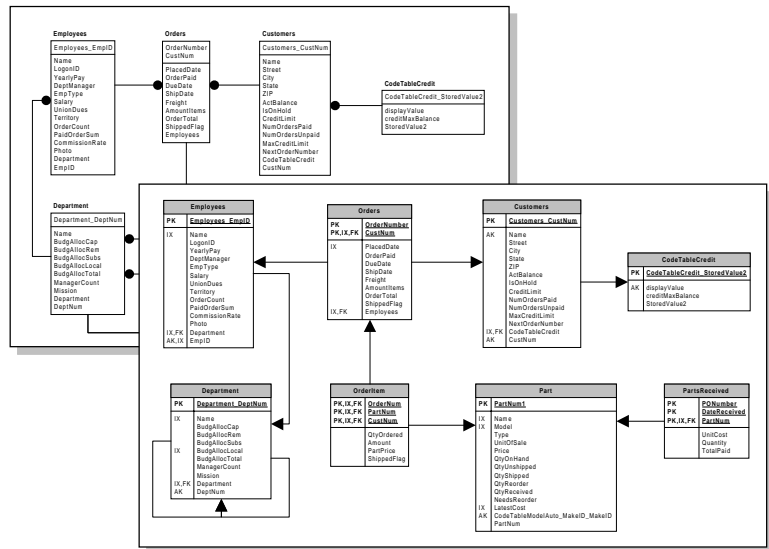
### To start a drawing with the Database Model solution:

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click Database, click Database Model.vst, then click OK.

For details about working in the Database Model solution, see the chapters in Part 3, “Modeling a Database with Visio Enterprise” later in this book.

**Figure 1-6**

You can reverse engineer a database to create models using IDEF1X or Relational notation with the Database Model template in Visio Enterprise.



## Installing the database diagram solutions

The database diagram solutions provide tools for creating notation-only database models. In the Visio Enterprise Setup program, if you choose the Database Diagrams option, you install the following templates:

- **Bachman** Includes shapes compliant with Bachman notation to use in creating entity relationship diagrams for data modeling in relational databases and information systems design.
- **Object Role Modeling** Includes shapes for objects (entities), constraints, connectors, predicates, and relationships for use in creating object-relational models and static diagrams in object-oriented design and analysis.

**NOTE** The Object Role Modeling template creates notation-only models. To create a conceptual model using Object Role Modeling (ORM) in the context of a database re-engineering project, you can install and work with VisioModeler. For details, see [“Installing VisioModeler”](#) below.

The database diagram solutions are installed in the \Visio\Solutions\Database folder.

### To start a drawing with a database diagram template:

- Start Visio Enterprise. In the Choose A Drawing Template dialog box, double-click Database, click the template you want, then click OK.

For details, choose Help > Template Help, double-click Visio Templates, double-click Database Diagram, then double-click the one you want.

## Installing VisioModeler

In the Visio Enterprise Setup program, the full version of the VisioModeler (formerly InfoModeler®) data modeling program is one of the Database installation options. VisioModeler is a database design, documentation, and analysis tool that allows you to manage every phase of the design process—from communicating with clients to generating a database schema.

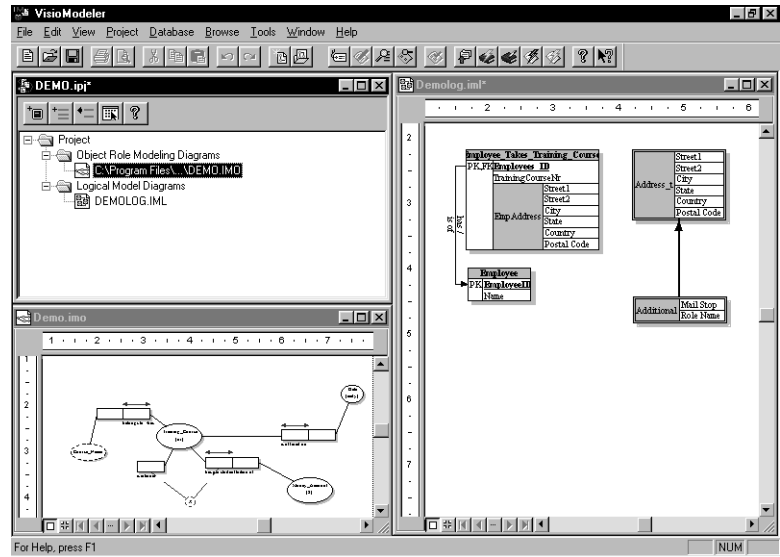
With VisioModeler, you can choose the modeling approach that suits your needs from these three approaches to database design: conceptual modeling using ORM, logical modeling using relational notation, and a mixture of logical and entity relationship modeling using IDEF1X notation. You can focus on capturing and expressing business information and examples in a source model document, and VisioModeler maps your conceptual model to a fully normalized logical model.

Certain types of models you create in VisioModeler can be imported in Visio Enterprise. For details, see [“Importing a data model”](#) in Chapter 7, [“Creating database models.”](#)

When you install VisioModeler, you have the option to install the user manuals *Guide to VisioModeler* and *Guide to FORML* in PDF format. These manuals are installed in the \Visio\VisioModeler\Manuals folder.

**Figure 1-7**

A model document in VisioModeler can contain a conceptual design of your application domain, from which you build a logical model.



## Installing ActiveQuery

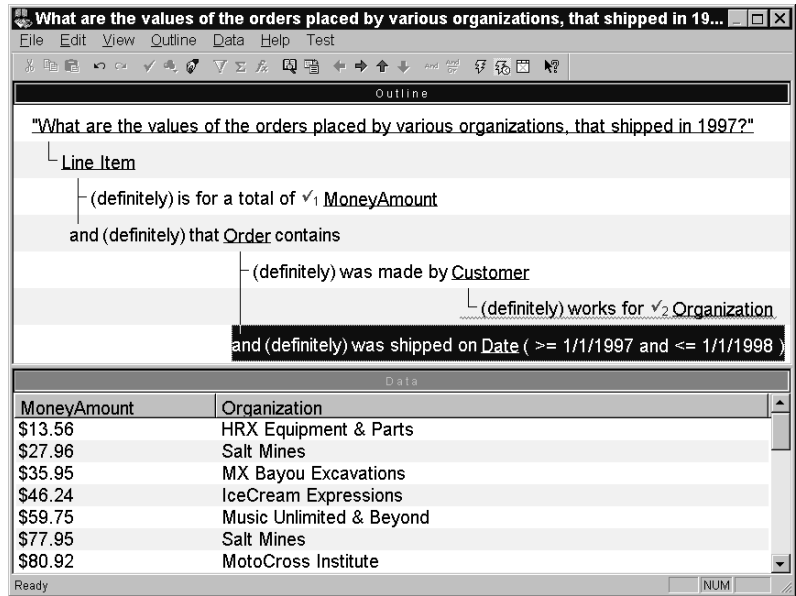
**NOTE** This program represents prototype technology that is not supported by Visio Technical Services.

In the Visio Enterprise Setup program, the full version of the ActiveQuery data retrieval, query, and analysis program is one of the Database installation options. Using the simple drag-and-drop interface in ActiveQuery, you can create and successively refine conceptual queries of a VisioModeler database model to produce exactly the results you want.

With ActiveQuery, you can translate raw database information from VisioModeler (.imd) files into natural language statements describing your query. You do not need to have specialized database knowledge to create a complete and accurate query. You select objects and facts from a Factbase to design a query using the innovative Point-to-Point and Query-by-Query technology. Then you can filter the data, apply aggregate functions, and specify a sort order to display data the way you want. You can even translate your query outline into an SQL statement that you can view and edit.

When you install ActiveQuery, you have the option to install the *Guide to ActiveQuery* user manual in PDF format. This manual is installed in the folder \Visio\ActiveQuery\ Manuals.

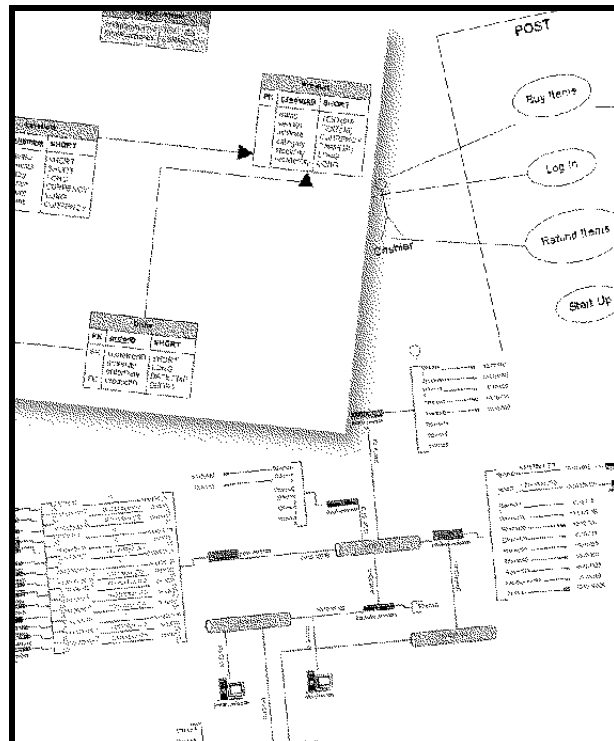
**Figure 1-8**  
Sample query showing a  
compound filter on *Date*;  
results appear in the Data View.



## Next steps

This book is designed in parts that correspond to the information technology areas that Visio Enterprise supports: network design and diagramming, software modeling, and database modeling. Your next step with Visio Enterprise depends on the type of diagram or model you are interested in creating.

<i>If you want to</i>	<i>See</i>
Learn the basics about working with Visio Enterprise shapes, stencils, and templates	<i>Using Visio Enterprise</i> or start the Visio Basics online tour
Use AutoDiscovery technology to diagram your network	<a href="#">Part 1, “Designing a Network with Visio Enterprise”</a>
Model software application architectures	<a href="#">Part 2, “Modeling Software with Visio Enterprise”</a>
Design and re-engineer databases	<a href="#">Part 3, “Modeling a Database with Visio Enterprise”</a>



## Designing a Network with Visio Enterprise



# Discovering and diagramming a network

Visio® Enterprise includes an AutoDiscovery And Layout solution that enables you to discover and diagram your entire network. The AutoDiscovery And Layout solution searches your network, gathers information about the devices on your network, and then creates a database with the information. You can customize discovery to include only specific networks or devices, or to discover every device on as many networks as you like.

Using the AutoDiscovery And Layout solution you can create a logical diagram of your network, compared to a network hardware diagram you can create with Visio Network Equipment (VNE). A logical diagram enables you to see the devices on your network, their connections, and their relationships.

You can create a single, large diagram to hang on a wall, or a diagram with several hyperlinked pages that “drill down” to show detailed information about selected network devices.

This chapter describes the Visio Enterprise AutoDiscovery And Layout solution. For details about VNE, see [Chapter 3, “Using network shapes.”](#)

## Topics in this chapter

- Discovering your network..... 24
- Creating a network diagram ..... 27
- Adding detail to a diagram using lists ..... 30
- Updating your diagram ..... 33
- Using VNE or other shapes..... 35
- Changing text colors for network devices ..... 40
- Managing the AutoDiscovery database..... 41
- Using multiple AutoDiscovery databases ..... 43









# Discovering your network

The AutoDiscovery™ technology in Visio Enterprise searches your network, polls devices, and gathers information about the devices, and then creates a database with the information. When you start the AutoDiscovery And Layout Solution, the Basic Discovery Wizard assists you in determining what kinds of network devices you want to discover and the router to use as a starting point. There is also an Advanced Discovery Wizard in which you can specify the networks and devices to include or exclude, and whether AutoDiscovery uses SNMP (Simple Network Management Protocol) or PING to find devices. Both wizards enable you to limit your search to only specific devices, or to discover every device on as many networks as you like.

### To open the AutoDiscovery And Layout template:

- Choose File > New > Network Diagram > AutoDiscovery And Layout.

The toolbar buttons for the AutoDiscovery And Layout template are shown and described below. They are available only when you open a network diagram in the AutoDiscovery And Layout template (ADL.vst). You can also right-click a device on the diagram to display a shortcut menu with additional context-sensitive commands, and access commands from the AutoDiscovery menu.

Button	Description
	<b>Basic Discovery</b> Starts the Basic Discovery Wizard used to discover your network.
	<b>Add A Network</b> Adds a network to the drawing to use as the starting point of your diagram.
	<b>Map Network Types</b> Changes the default shapes used for network types.
	<b>Map Device Types</b> Changes the default shapes used for specific devices or device types.
	<b>Map Text Colors</b> Changes the text colors used on items added to a diagram.
	<b>Guided Update</b> Starts the Guided Updating™ feature and displays a window that lists the devices, interfaces, and networks that have been added to, or changed on, your network.
	<b>Database Viewer</b> Starts the Database Viewer, where you can delete and rename objects in the database, set the CIR (Committed Information Rate) for DLCIs (Data Link Connection Identifiers), and set interface speeds.
	<b>Network Diagram Assistant</b> Accesses the Network Diagramming Assistant for help on what to do next.



## How long does discovery take?

No two networks are alike, so it is difficult to specify how long it will take to discover your network. It can go as quickly as a few minutes for a small network, or up to a number of hours for a large network. For example, it may take overnight to discover a network with 200 routers. The following factors can help you determine how long it can take to discover a network:

- **The number of device types discovery looks for** The more options you select in the Advanced Discovery Wizard, such as types of devices to be discovered and the number of specified discovery domains, the longer it will take to discover.
- **The PING and SNMP timeout settings** You can specify how long discovery should wait before timing out if a device doesn't respond to a PING or SNMP request. For example, if you have many devices that don't respond to SNMP requests, you may want to decrease the SNMP timeout field.
- **Link speeds** The types of links used in your environment will affect how long discovery takes. WAN (wide area network) links take longer to discover than LAN (local area network) links. For example, if you have a remote office connected by a WAN link, the packet exchange will be slower. You may want to exclude the devices on remote subnets from your discovery domain. On the other hand, for this scenario you might want to increase the number of SNMP retries as well as SNMP and PING timeout settings.

## Using the Basic Discovery Wizard

The Basic Discovery Wizard allows you to determine what kind of network is discovered, a Routed network or a Switched/Bridge network. You can also specify any SNMP community strings used throughout your network, and specify whether to use the default gateway, or specify another router to use as the starting point for discovery. In general, if you are unsure about what to specify on a wizard screen, accepting the default will enable the most discovery of your network. If some devices are not found, you can use the Advanced Discovery Wizard to discover those devices.

### **To discover network devices with the Basic Discovery Wizard:**

- 1 Choose File > New > Network Diagram > AutoDiscovery And Layout.

The AutoDiscovery And Layout template opens.



- 2 On the AutoDiscovery And Layout toolbar, click the Basic Discovery button to begin working in the Basic Discovery Wizard and discover your network.
- 3 In the wizard, answer the questions on each screen, or press Next on each screen to accept the default settings. The default search uses SNMP to discover those devices that are active on your network and have SNMP running.

When discovery begins, the Discovery Monitor opens to show the progress of your network search. When discovery is finished, the message “AutoDiscovery is Finished” is displayed in the Discovery Monitor window, and the AutoDiscovery database is created. You can now begin adding networks and devices to your network diagram.

## **Using the Advanced Discovery Wizard**

The Advanced Discovery Wizard enables you to control more of what the AutoDiscovery technology discovers and how it gathers information. You can specifically include and exclude networks and devices, specify whether discovery uses SNMP or PING to determine what devices are on the network, and determine the order that SNMP community strings use.

### **To discover network devices with the Advanced Discovery Wizard:**

- 1 Choose File > New > Network Diagram > AutoDiscovery And Layout.

The AutoDiscovery And Layout template opens.

- 2 Choose AutoDiscovery > Discover > Advanced Discovery to begin working in the Advanced Discovery Wizard.
- 3 Answer the questions on the wizard screens.

For details about the options on a screen, click the Help button on that screen.

When you have completed the last wizard screen, the Discovery Monitor opens to show you the progress of the discovery of your network. When discovery is finished, the message “AutoDiscovery is Finished” is displayed in the Discovery Monitor window, and the AutoDiscovery database is created. You can now begin adding networks and devices to your network diagram.

# Creating a network diagram

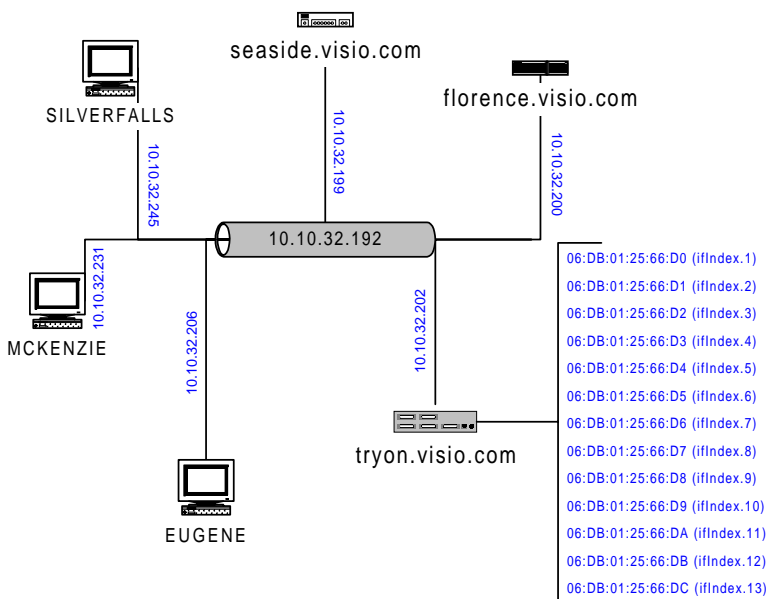
Before you create a network diagram from the information returned during discovery, consider the kind of diagram you want and how you plan to use it. Do you want a single network diagram to hang on your wall, showing all devices on your entire network? Or do you want several smaller diagrams that “drill down” into your network, linked together and published on your company’s intranet? How many backbone networks does your company have? Do you want a separate diagram for each geographical region?

Visio Enterprise enables you to create a one-page network diagram, or a hyper-linked diagram you can publish on the Web. In either case, you can make the diagram as simple or complex as your network requires.

## Creating a one-page diagram

A one-page diagram is useful when you want to show your network on a single diagram. The page size you choose for a one-page diagram is important because you want to choose a size that is large enough to accommodate all of your network devices and detail. Once completed, one-page diagrams are useful for hanging on a wall, where you can easily see your network layout and connectivity.

**Figure 2-1**  
Example of a one-page network diagram, including an interface list



### To create a one-page diagram:

- 1 Choose File > New > Network Diagram > AutoDiscovery And Layout.

The AutoDiscovery And Layout template opens.

- 2 To specify the page size of your diagram, choose File > Page Setup, and on the Page Size tab, select the size you want.

The ANSI Engineering or ANSI Architectural sizes are useful for large diagrams.



- 3 On the AutoDiscovery And Layout toolbar, click the Add A Network button.
- 4 In the Select A Network To Add dialog box, select a network, then click OK.

**NOTE** You must have previously used the Basic Discovery Wizard or Advanced Discovery Wizard for networks to appear in this dialog box.

Usually you begin by adding your backbone network to the diagram.

A shape representing the network is automatically added to the diagram.

- 5 To connect network devices, such as routers, to the network, right-click the network in the diagram, then choose Connect Devices from the shortcut menu.
- 6 In the Connect Devices dialog box, select those devices connected to the network that you want to place on the diagram. Check Attach Interface IP Address To Links to display the IP address of links between two devices on the diagram.

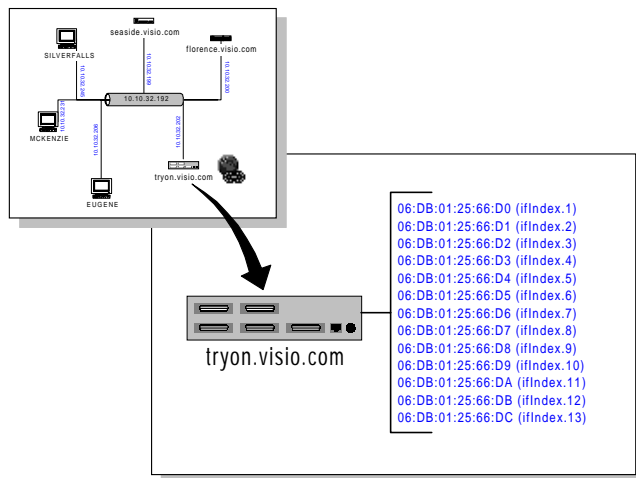
You can continue to add networks and devices to your diagram. You can also add lists to your diagram. For details about lists, see [“Adding detail to a diagram using lists”](#) later in this chapter.

## Creating a hyperlinked diagram

You can create a hyperlinked diagram that is composed of many small diagrams representing your network, which you can place on your company’s intranet or distribute to a specific group of people. For example, you can create an overview page of all your networks, and then create hyperlinks from each network to “drill down” into that network and see connected devices and details about them. This allows you to diagram an overview of your entire enterprise network, and then view only those parts in which you are interested.

**Figure 2-2**

You can place hyperlinks between devices so that you can link diagrams together and “drill down” into your network.



### To create a hyperlinked diagram:

- 1 Choose File > New > Network Diagram > AutoDiscovery And Layout.

The AutoDiscovery And Layout template opens.



- 2 Click the Add A Network button, and in the dialog box, select a network, then click OK.

**NOTE** You must have previously used the Basic Discovery Wizard or Advanced Discovery Wizard for networks to appear in this dialog box.

Usually you begin by adding your backbone network to the diagram. A shape representing the network is automatically added to the diagram.

- 3 To connect network devices, such as routers, to the network, right-click the network shape, then choose Connect Devices from the shortcut menu.
- 4 In the Connect Devices dialog box, select those devices you want to place on the diagram and connect to the network. Check Attach Interface IP Address To Links to display the IP address of links between two devices on the diagram.
- 5 Continue adding networks and connecting devices to them.
- 6 To create a hyperlink for a network or device, right-click the network or device, then select Create Hyperlinked Page from the shortcut menu.

Visio Enterprise opens a new page and creates a hyperlink from the network or device to the new page. A link from the new page back to the original page is also automatically created so that users can navigate back to the original diagram.

- 7 On the new page, continue adding and connecting devices.

You can create hyperlinks from this page to create another page that shows even more detail about the network device.

# Adding detail to a diagram using lists

After you complete a basic diagram that includes a few networks and devices, you can add more detail. Listing objects on your diagram helps organize the network objects without cluttering your diagram. You can create a list of devices, device counters, DLCIs, and interfaces, and place that list on your diagram next to the selected object to which the list applies. You can right-click a network device to see the list commands available on the shortcut menu. The type of list commands available depends on the type of selected network device. For example, to create a list of DLCIs, you must select a frame relay device.

You can combine lists with objects, and create multiple lists for the same device. You can create one long list, or multiple smaller lists that are hyperlinked together on Web-based diagrams.

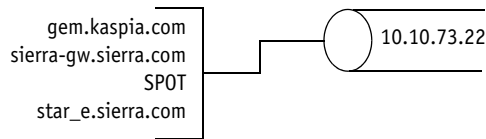
You can use lists in the following ways:

- **Combine lists with objects** You can use lists to decrease the number of objects on your page. You may want to diagram the connectivity objects on your network and then use a list to display the workstation names. You can also create hyperlinks to other pages.
- **Create multiple lists for the same device** When you choose a List command, a dialog box displays the items you can include on the list. If you want to draw a list of all the items attached to a device, you can create several lists rather than one large one. You can then rearrange the lists on the page so that they don't run off the page or crowd other objects.
- **Long versus short lists** Long lists may be more appropriate for a wall-sized chart because they will fit better on the larger paper sizes used for wall diagrams (usually ANSI Engineering size option D or E). For smaller paper sizes, you may want to create multiple lists, or hyperlink separate diagrams together.

## Device lists

A **device list** shows all of the devices that are connected to the selected network or device. You can use this type of list if you have a complex network and want to limit the number of devices on your diagram so that it is easy to read. In this case, you might diagram your most important devices, and then use device lists to see all devices that are connected to the network.

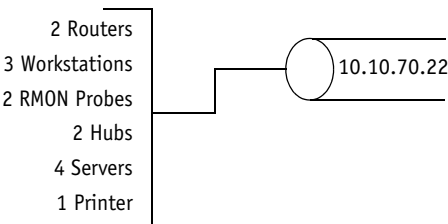
**Figure 2-3**  
Sample device list



## Device counter lists

A **device counter list** shows all device types and the quantity of each type that is connected to the selected network. You can use this type of list to show you exactly what is connected to the selected network. If you need to redesign your network's layout, you can use a device counter list to help determine potential traffic patterns on your network and how to decrease high-traffic spots.

**Figure 2-4**  
Sample device counter list showing the types of devices connected to the network 10.10.70.22



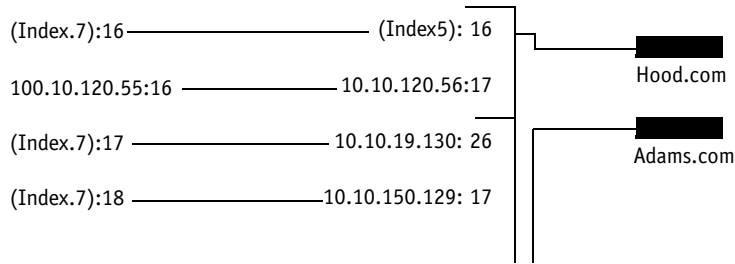
## DLCI lists

A **DLCI list** shows the DLCIs that are connected to the selected device. Using a DLCI list you can easily document your frame relay circuits. The AutoDiscovery And Layout solution lists the DLCIs connected to a device and automatically connects the DLCI endpoints.

The AutoDiscovery And Layout solution discovers all DLCI endpoints; however, it cannot correlate both endpoints for each DLCI. It resolves one endpoint for each DLCI. You can use the Create Circuits command to specify the other endpoint for each DLCI. Access the Create Circuits command by right-clicking a frame relay router and selecting the command from the shortcut menu.

**TIP** If you have displayed connection points in your diagram (View > Connection Points), you may want to turn off their display so that the diagram is easier to read.

**Figure 2-5**  
Sample DLCI list showing the DLCIs connected to the router Hood

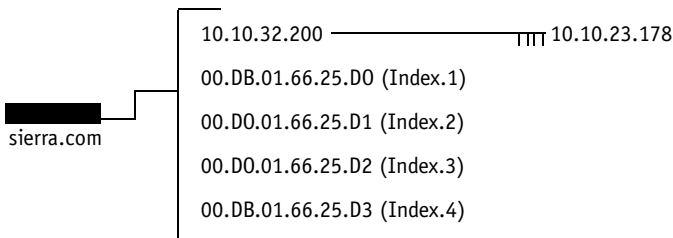


**NOTE** Frame relay support is available if you have purchased and installed the Frame Relay Add-on.

## Interface lists

An **interface list** is a list of the interfaces that are connected to the selected device. You can use an interface list as a troubleshooting tool to determine to which networks your interfaces are connected. When you add an interface list to your diagram, you can choose to add the networks to which the interfaces are connected.

**Figure 2-6**  
Sample interface list showing the interfaces connected to the device sierra.com



### To add a list to a diagram:

- 1 On the diagram, right-click the network device to which you want to add a list, then choose the type of list to create from the shortcut menu.  
The list commands that are available depend on the type of device you selected. For example, you cannot add a DLCI list to a device that is not a frame relay device.
- 2 In the dialog box that appears, select the items to include in the list. Also select whether the list is placed to the left or right of the device. You can drag the list to a different location after placing it on the diagram.
- 3 Click OK to close the dialog box and place the list on the diagram.
- 4 To create multiple lists, repeat these steps.

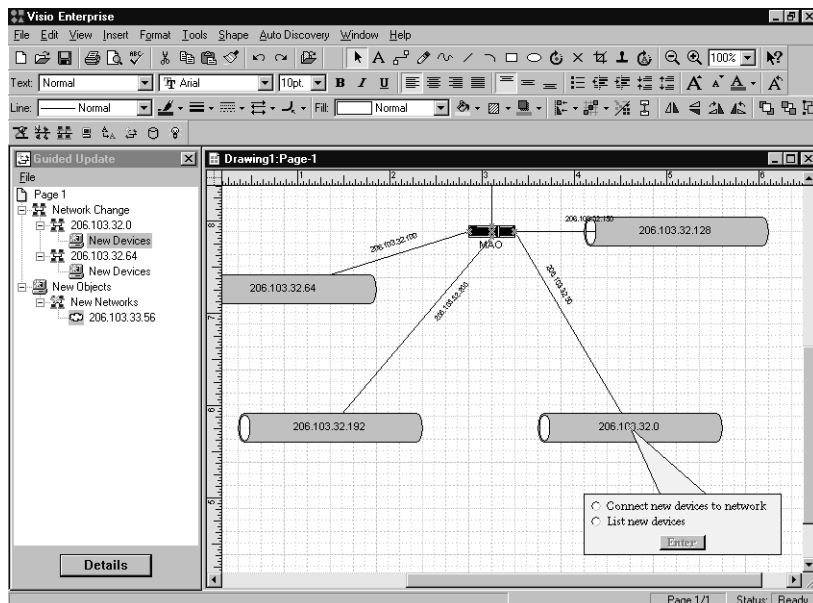


# Updating your diagram

You can easily view and incorporate network changes using the Guided Update command. Use the Guided Update command after you rerun discovery to display a list of the components on your network that were added, modified, or deleted since the last time you discovered the network. The changes are displayed in the Guided Update window in a tree view.

**Figure 2-7**

Sample Guided Update window showing the network changes and the available tasks



The Guided Update command shows the following changes:

- **Device Changes** Shows new or deleted interfaces.
- **Network Changes** Shows devices added to or deleted from the network.
- **New Objects** Shows new network objects.

When the changes appear in the Guided Update window, you can do the following:

- Apply them to your diagram immediately.
- Save the changes to a file to apply to your diagram another time. Do this if you want to save the changes but not put them on the diagram at this time. The next time you run Guided Update, the new changes display in the Guided Update window, and the previous changes are overwritten. You can then open the saved file to apply the previous changes.
- Load a previously saved file of network changes.

**NOTE** Use Guided Update after you rerun discovery on your network. Guided Update displays a list of the components on your network that were added, modified, or changed since the last time you discovered the network.



**To update your diagram:**

- 1 Click the Guided Update button on the AutoDiscovery And Layout toolbar.
- 2 Expand the type of change you want to see: device changes, network changes, or new objects.
- 3 Select the specific device or change you want more information about, then click Details.

The Details window shows the details about what changed. For example, if you select a device under a specific network in the Network Changes list, then click Details, the Details window shows the devices added to the network.

- 4 The diagram displays the actions available for the selected entry to guide you in what you can do next. Choose the button for the action you want to perform, then click Enter. The appropriate dialog box appears in which you can update the diagram with the change.

The action may remain on the page if you perform only part of the available changes. For example, if several devices were added to a network and you added only some of them to the diagram, the action remains on the page because you can add more devices to the page. To remove the action from the drawing, deselect the item in the Guided Update window.

If you do not want to incorporate all the changes at one time, you can save the list, open it later, and apply more changes. The next time you run Guided Update, the new changes display in the Guided Update window, and the previous changes are overwritten. You can then open the saved file to apply the previous changes.

**To save a list of changes:**

- 1 Choose File > Save from the Guided Update window to save the list of changes.
- 2 To open the file at a later time use File > Open from the Guided Update window and navigate to the file.

## Using VNE or other shapes

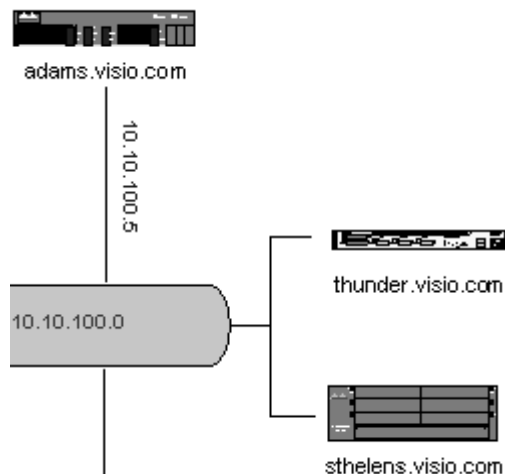
The AutoDiscovery And Layout solution uses default shapes that are assigned to network objects by matching the SysObjectID of the device to a shape found in the Visio Enterprise network stencils. A SysObjectID is a value obtained using SNMP that vendors use to identify their products.

The AutoDiscovery And Layout solution provides a shape mapping file that maps network device types to correct VNE shapes. The solution uses this file to determine the network shapes that are displayed on your diagrams. This file is continually updated with new stencils and is available with VNE. If a specific VNE shape is not available for a discovered object, a default shape is used.

You can also use shapes that you have created or shapes from other stencils. The shapes you specify to use for network and device mappings take precedence over the AutoDiscovery And Layout solution's default mappings.

**Figure 2-8**

In this diagram, VNE network shapes are used instead of the default template shapes.



**NOTE** When you change the default shape mapping of a device type or network object, the new mapping is used the next time that device is imported to your diagram. It does not affect shapes currently on a diagram.



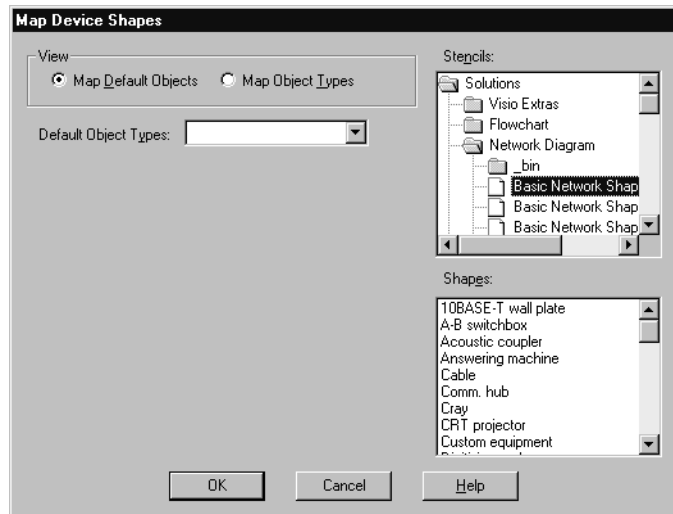
### To use other device shapes instead of the template default shapes:

- 1 Click the Map Device Shapes button on the AutoDiscovery And Layout toolbar.
- 2 In the Map Device Shapes dialog box, choose Map Default Objects.
- 3 From the Default Object Types list, choose a device type.
- 4 Under Stencils, locate the stencil (.vss file) you want to use.
- 5 Under Shapes, select the shape to which you want to map the device type.  
This mapping is temporarily saved when you select a shape. You can continue to map different default objects before closing the dialog box.
- 6 Click OK.

All of the mappings you selected are saved. The selected shapes are used the next time you add the selected device type to your diagram.

**Figure 2-9**

Map Device Shapes dialog box with Map Default Objects selected





### To use other device shapes for specific devices:

- 1 Click the Map Device Shapes button on the AutoDiscovery And Layout toolbar.
- 2 In the Map Device Shapes dialog box, choose Map Object Types.
- 3 Check the device types you want to view in the list box, then click Load Objects.
- 4 Under Name, select the device you want to map.
- 5 Under Stencils, locate the stencil (.vss file) you want to use.
- 6 Under Shapes, select the shape to which you want to map the device.

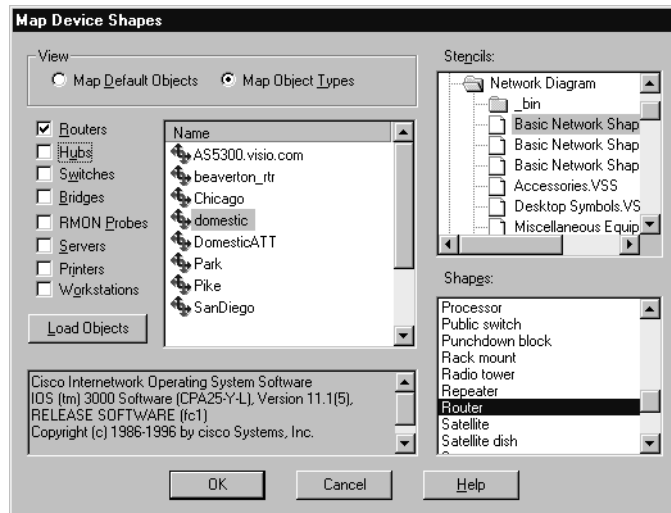
This mapping is temporarily saved when you select a shape. You can continue to map different default objects before closing the dialog box.

- 7 Click OK.

All of the mappings you selected are saved. The selected shape is used the next time you add the selected device to your diagram.

**Figure 2-10**

Map Device Shapes dialog box with Map Object Types selected





### To use other network shapes:

- 1 Click the Map Network Shapes button on the AutoDiscovery And Layout toolbar.
- 2 In the Map Network Shapes dialog box, choose Large Shapes or Small Shapes.

To change the shape used for networks placed on the diagram, select Large Shapes. To change the shape used for network icons used in interface lists, choose Small Shapes.

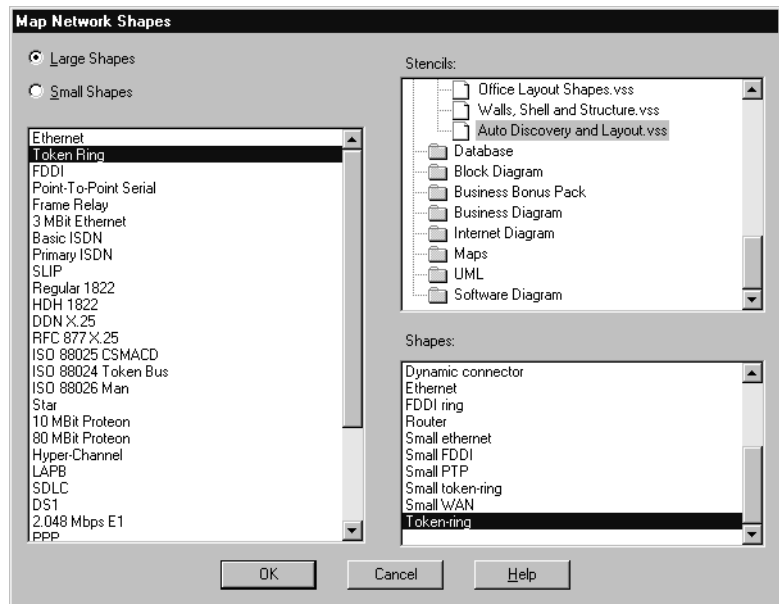
- 3 In the left pane, select the network type you want to map.
- 4 Under Stencils, locate the stencil (.vss file) you want to use.  
The shapes contained in the stencil appear in the Shapes pane.
- 5 In the Shapes pane, select the shape to map to the network type.

This mapping is temporarily saved when you select a shape. You can continue to map different network types before closing the dialog box.

- 6 Click OK.

All of the mappings you selected are saved. The selected shape can be used the next time you add the selected network type to your diagram.

**Figure 2-11**  
Map Network Shapes dialog box



If you have other shapes that you want to use in your diagrams, you can customize Visio Enterprise to display them in the Map Network Shapes dialog box.

**To use other stencils and shapes:**

- 1 Choose Tools > Options, then click the File Paths tab.
- 2 In the Stencils field, add a semicolon (;) at the end of the existing path, followed by the path to your stencils.
- 3 Click OK.

These shapes will be available the next time you open the Map Network Shapes dialog box.

If you have created custom shapes and decide you want to use the default ones instead, you can delete the custom shapes files. Returning to the default shapes will not affect any diagrams you have already created, but the default shapes will be used the next time you import a network object to your diagram.

**To delete custom shapes and use the default shapes:**

- From the Visio\Solutions\Network Diagram\\_bin folder, where Visio Enterprise program files are installed, delete the following files:

UserDeviceMap.txt

UserNetMap.txt

If you decide to create custom shapes again in the future, these files will be re-created with the new shapes.

## Changing text colors for network devices

You can change the default text colors used for various network devices and choose the color scheme you want to use. For example, you can make all workstation names or addresses one color, printers a different color, and routers another color. This allows you to easily see the different types of devices on your diagram, and to color-code them.

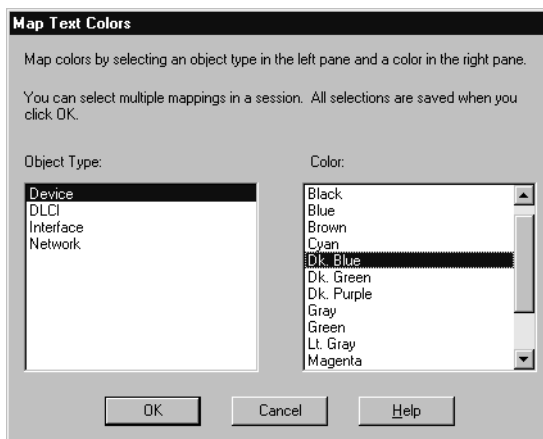
**NOTE** The colors you change will not be applied to existing devices on a diagram. They will be used the next time you add a device to the diagram.

### To change text colors:

- 1 Click the Map Text Colors button on the AutoDiscovery And Layout toolbar.
- 2 In the Map Text Colors dialog box, under Object Type, choose an object type.
- 3 Under Color, choose a color to which to map the object type.  
You can select multiple mappings in a session.
- 4 Click OK to save the new mappings.



**Figure 2-12**  
Map Text Colors dialog box

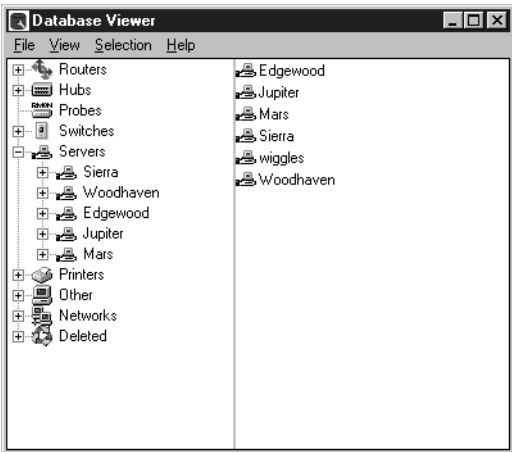




# Managing the AutoDiscovery database

When you use the AutoDiscovery And Layout solution, Visio Enterprise places the information about the types of devices on your network in the AutoDiscovery database. You can view the contents of the database with the Database Viewer. You can also use the Database Viewer to delete (and undelete) entries from the database, rename a device in the database, change the speed of an interface, and change the CIR of a frame relay circuit.

**Figure 2-13**  
Database Viewer tree view of the contents in the AutoDiscovery database



**NOTE** For details about a specific device, such as the manufacturer, SNMP information, or IP address, right-click the device in the diagram, then choose Device Properties from the shortcut menu.

**To open the Database Viewer:**



- Click the Database Viewer button on the AutoDiscovery And Layout toolbar.

**To delete entries from the AutoDiscovery database:**

- 1 In the Database Viewer, expand the object types and select the device(s) you want to delete. To select multiple devices in a contiguous list, select the first device, press and hold the Shift key, then select the last device in the list. To delete devices not in a contiguous list, select the first device, press and hold the Ctrl key, then select the remaining devices.

If you delete a device that has associated interfaces, the interfaces are also deleted. For example, deleting a router deletes all of its interfaces as well.

- 2 Choose Selection > Delete.

The device(s) is placed in the Deleted group. The Deleted group is like a recycle bin. Once a device is in the Deleted group, you can leave it there, or you can permanently delete the device from the database. You can also undelete a deleted device from the Deleted group and place it back in the tree view.

- 3 To permanently delete the device, open the Deleted group and select the device. Then choose Selection > Delete.

**NOTE** Choosing to delete devices from the AutoDiscovery database permanently does not decrease the size of the database. When new devices are discovered and added to the database, the database size does not increase until more devices are added than have been deleted.

If you permanently remove an object by mistake, the AutoDiscovery technology will rediscover it the next time you run discovery (assuming it is still accessible). This is useful if you remove an object that is actually still part of the network.

**To undelete entries marked for deletion:**

- In the Database Viewer, select the device from the Deleted group, then choose Selection > Undelete.

The device is placed in the group from which it was deleted.

**To rename a device currently in the AutoDiscovery database:**

- 1 In the Database Viewer, select the device whose name you want to change.
- 2 Choose Selection > Rename.
- 3 In the Rename Object dialog box, enter the new name, up to 255 characters.
- 4 Click OK.

**To change the speed of an interface:**

- 1 In the Database Viewer, select the interface whose speed you want to change.
- 2 Choose Selection > Change Speed.
- 3 In the Set Interface Speed dialog box, enter the interface speed in the Speed (bps) field.
- 4 Click OK.

**To change the CIR of a frame relay circuit:**

*Frame relay support is available if you have purchased and installed the Frame Relay Add-on.*

- 1 In the Database Viewer, select a DLCI whose CIR you want to change.  
Typically, the CIR stored in the frame relay MIB (management information base) for a virtual circuit is 0. This command allows you to set the CIR to its correct value.
- 2 Choose Selection > Change CIR.
- 3 In the Edit DLCI CIR dialog box, select one of the predefined CIR settings, or choose Custom to enter a setting of your own.
- 4 Click OK.

## Using multiple AutoDiscovery databases

You can create and use multiple AutoDiscovery databases and create different network diagrams that are not related to each other for each database. For example, field service engineers may want to diagram several customers' networks and keep each customer's network information in a separate database. Then when making customer visits, one could open the appropriate database. Or for very large networks, you may want to use the Advanced Discovery Wizard to discover a specific part of a network and create one database, and then discover another part of the network and create a different database. You can then diagram specific portions of the network easily without having to wade through the entire network data. This enables you to efficiently discover, diagram, and update portions of a large network.

Databases do not have a specific name because they are made up of several files that reside in a directory. When creating a new database, you specify the directory where the files are placed. Each database must be in a unique directory; you cannot place two databases in the same directory.

When you specify a new database, be sure to place it in a descriptive directory so that you can determine what type of information is in the database. For example, you may create a database for Company A and a different database for Company B. You could place Company A's database in a directory named `\company_a`, and likewise, Company B's database in `\company_b`.

**To create a new database:**

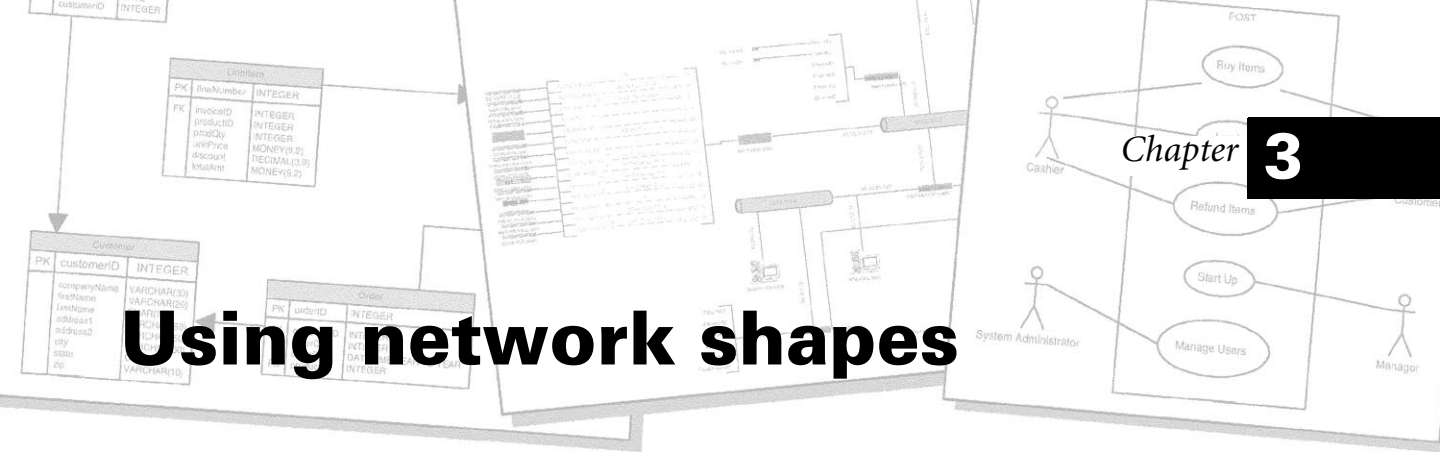
- 1 Choose AutoDiscovery > Configuration > Change Database Directory.
- 2 In the dialog box, enter the location where you want to create a new database.
- 3 Click OK.
- 4 Restart your system to restart the database software.

The next time you run discovery, the network information is placed in the new database.

When you're working with multiple databases, the database you use to create a diagram must be the active database when working with that diagram. If you open a diagram and do not have the database from which it was created open, you are prompted to open that database before continuing. You cannot edit or change the diagram until you have the appropriate database specified. You can, however, print the diagram regardless of the current database.

**To specify a database to use:**

- 1 Choose AutoDiscovery > Configuration > Change Database Directory.
- 2 Enter the location of the database to use.
- 3 Click OK.
- 4 Restart your system to restart the database software and use the database you specified.



## Chapter 3

# Using network shapes

Visio® Enterprise includes Visio Network Equipment (VNE), which includes thousands of manufacturer-specific network shapes designed to help you plan and document a physical network diagram. Using these shapes with the powerful diagramming features of Visio Enterprise, you can produce physical network diagrams that are complete and accurate down to the last detail. You can

- Mount exactly scaled network components on racks.
- Fit network modules into the corresponding network chassis.
- Attach data to shapes and generate reports on costs, inventories, and other custom properties that you define.
- Show levels of a network, from a wide area network (WAN) to a wiring closet, with links to jump from one level to another.

### Topics in this chapter

- Working with network shapes ..... 46
- Creating a physical network diagram ..... 49
- Working with labels ..... 51
- Connecting 2-D shapes ..... 52
- Creating reports ..... 53
- Modifying a network shape and using the VNE Property Duplicator ..... 54

# Working with network shapes

Here are a few basic facts about VNE shapes to help you work with and use them efficiently.

**For best results, start with a VNE template.** Most network shapes are designed to work best with a 1:10 page scale, the setting used in these templates. The exceptions are monitors and workstations, which use a 1:16 scale, and very small shapes, such as faceplates and bezels, which use a 1:6 scale.

## **Search for specific network equipment shapes using these methods:**

- Use the **Locate Network Equipment utility**, which opens automatically when you start a drawing with a VNE template. You can browse for shapes grouped by property, quickly search for a specific shape, or type a keyword, such as “*router*,” to see all the shapes associated with that keyword, or select shapes of only the particular manufacturers or equipment types you want. You can move and detach the Locate window.

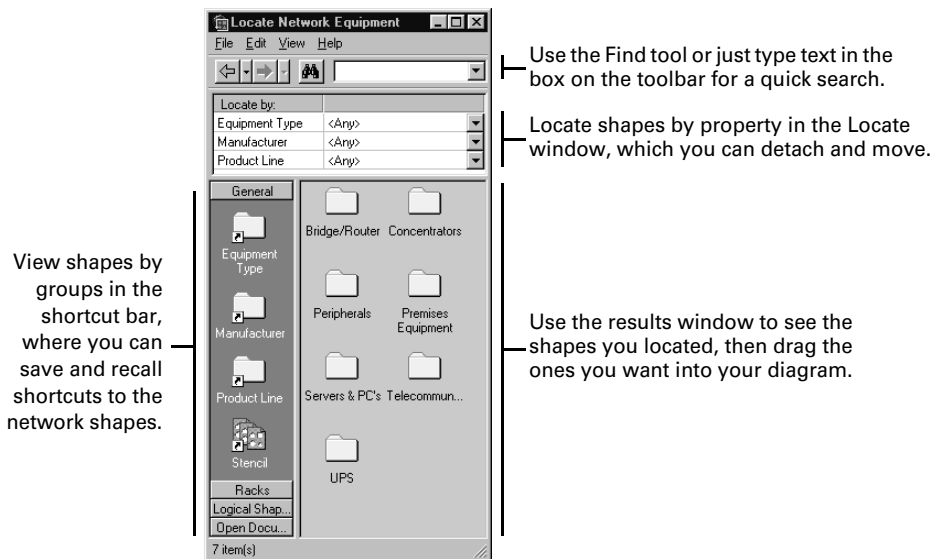
The Locate Network Equipment utility replaces the Visio stencil window with a shape database manager that allows you to quickly locate a specific network shape among the thousands available.

**NOTE** You can locate only VNE shapes with the Locate Network Equipment utility. It does not locate other Visio Enterprise shapes.

- Look up shapes in the **Visio Network Equipment Directory**, which lists all VNE shapes by manufacturer and product line. For details, see [“Using the Visio Network Equipment Directory”](#) later in this chapter.
- Search directly through open stencils.
- Use keyword searches to find individual shapes. Use category searches to find groups of shapes.

You can also modify network shapes saved in Windows Metafile format; however, you must first ungroup them. For details, see [“Modifying a network shape and using the VNE Property Duplicator”](#) later in this chapter.

**Figure 3-1**  
The Locate Network  
Equipment utility



**Get online help when you need it.** As you work with the VNE templates, stencils, and shapes, you can easily get the online help you need in these ways:

- For help about VNE shapes, right-click a network shape on the drawing page, then choose Shape Help from the shortcut menu.
- For help about the Locate Network Equipment utility, use the Help menu in the utility's menu bar.
- For general help about templates and stencils, choose Help > Visio Help, click Index, then type "*templates*" or "*stencils*."

## Using the Locate Network Equipment utility

The Locate Network Equipment utility gives you several quick ways to find equipment shapes. Instead of opening a stencil containing the shape you want, you can use the Locate Network Equipment utility to find a particular shape or several shapes that meet criteria you set.

The utility searches for VNE shapes on your hard disk and optionally on the Visio Network Equipment CD. The resulting shapes are displayed in the utility's results window, and you can then drag a shape directly into your diagram.

The Locate Network Equipment utility starts automatically when you open a VNE template. If you open a diagram created in an earlier version of Visio Network Equipment, or open with a blank drawing page, you must start the utility manually.

**To open the Locate Network Equipment utility:**

- To open a new file and the Locate Network Equipment utility, choose File > New > VNE > Closet, Lan, or Wan.
- To open the Locate Network Equipment utility from an existing file, choose Tools > Macro > VNE > Locate Network Equipment.

**To locate equipment by category:**

- 1 Use the Locate By fields in the Locate window to display equipment by type, manufacturer, product line, or a combination. You can move and detach the Locate window.
- 2 Click the shortcuts in the General group to display equipment by type, manufacturer, product line, or stencil.
- 3 Click the shortcuts in the Racks group to display racks.
- 4 Click the shortcuts in the Logical Shapes group to display connectors, call-outs, and other tools.

**To find an equipment shape using the keyboard:**

- 1 Click the Find button on the Locate Network Equipment toolbar, or choose Edit > Find in the Locate window.
- 2 Type a name, keyword, or product number in the Find What text box on the toolbar, then click Find or press Enter.

**NOTE** The search range covers the properties selected in the Find dialog box.

**To interrupt a search:**

- Click the Stop button on the toolbar. The results window displays the search results up to that point.

## Using the Visio Network Equipment Directory

The Visio Network Equipment Directory lists all VNE shapes by manufacturer and product line. For every shape, the VNE Directory shows the product ID, product description, and stencil file name.

**To open the Visio Network Equipment Directory:**

- Right-click a network equipment shape in the diagram, then choose Shape Help from the shortcut menu. On the Contents screen, you can choose to view the Visio Network Equipment Directory by clicking View The VNE Directory.
- If you are using a VNE template, choose Help > VNE Shape Directory to open the Visio Network Equipment Directory.



**To find a shape in the VNE Directory:**

- Click Contents to open a manufacturer and see a list of its product lines, then click on a product line to see the VNE shape list. For a manufacturer's complete shape listing, choose All Shapes.
- Click Index to search by the manufacturer or product line keyword, then jump quickly to a particular product line.
- Click Find to use the full-text Find feature to search through the entire VNE Directory for any text you want to find.

## Creating a physical network diagram

VNE shapes are precisely scaled to fit together, and their connection points make it easy to snap them into place. Once together, chassis, racks, and modules remain joined together, even when they're moved. You can use these features in creating detailed physical network diagrams.

**To create a physical network diagram:**

- 1 Choose File > New > VNE > Closet, Lan, or Wan.

When you start a diagram with one of these templates, the correct drawing scale is set for you so that shapes work properly, and the Locate Network Equipment utility opens automatically.

- 2 Use the Locate Network Equipment utility to locate the shapes you want.
- 3 Drag the shapes you want from the results window and drop them onto the drawing page.
- 4 Arrange the shapes to represent the network layout you want.
- 5 Add any necessary connections between network shapes.

### To put a chassis in a rack:

- 1 In the Locate Network Equipment window, click the Tools Racks group, then click Racks And Connectors.
- 2 Locate the rack you want, then drag it from the results window onto the drawing page.
- 3 Locate the chassis shape you want, then drag it in position over the rack.
- 4 Drop the chassis, snapping it onto the rack. Connection points at the lower corners of the chassis glue to the connection points (x) on the rack.

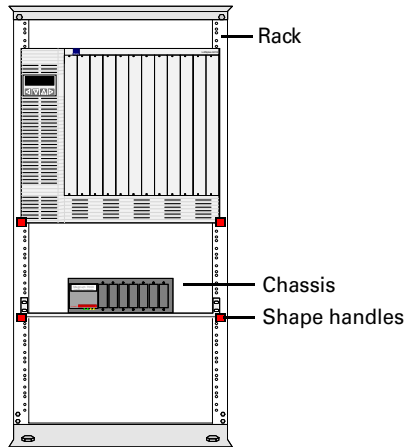
All rackmountable shapes fit into the VNE racks. Chassis and other rack-mounted devices have a mounting frame for precise placement.

- 5 If needed, drag the chassis up or down to adjust it vertically. Rack connection points (x) are spaced at screwholes on the rack.

Use the same procedure to attach shelves to racks to hold smaller pieces of equipment.

**Figure 3-2**

When you glue a chassis to a rack, shape handles turn red.



### To put a module in a chassis:

- 1 Choose Tools > Macro > VNE > Locate Network Equipment to open the Locate Network Equipment window.
- 2 In the Locate Network Equipment window, locate the chassis shape you want, then drag it onto the drawing page.
- 3 Locate the module you want, then drag it into its slot on the chassis.

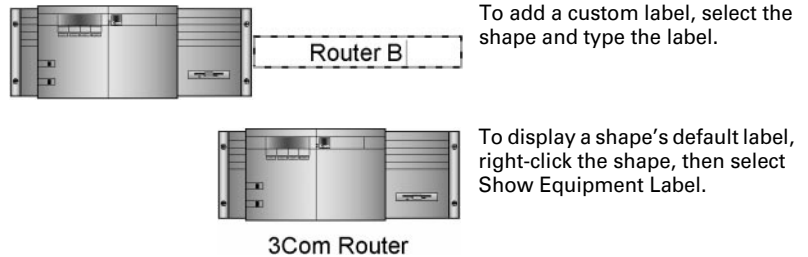
If the module's orientation (horizontal or vertical) does not match the slot, rotate the module. To do this, select the module, then choose Shape > Rotate Left or Shape > Rotate Right.

When the module is properly connected to the slot, connection points at each end of the module turn red, indicating that they are glued.

# Working with labels

You can add custom labels and display equipment labels on VNE shapes. By default, label text is 8-point Arial, with centered paragraph alignment. Use commands on the Format menu to make format changes.

**Figure 3-3**  
Adding and displaying labels



## To add a custom label to a shape:

- Select the shape, then type the new label. The text appears in a text block to the right of the shape.

## To reposition a custom label (text block):

- Right-click the shape, then choose Show Text Control Handle from the shortcut menu. Drag the control handle to the position you want.

## To remove a custom label:

- Click the text tool, then click in the text block and delete it.

## To display a shape's default label:

- Right-click the shape, then choose Show Equipment Label from the shortcut menu. The manufacturer name and product number appear in a text block near the shape.

## To hide an equipment label:

- Right-click the shape, then choose Hide Equipment Label from the shortcut menu.

## To reposition an equipment label:

- Drag the label's control handle.

A

# Connecting 2-D shapes

Use the connector tool to draw a right-angled connector between 2-D shapes. You can also use special connector shapes to draw other connector types.

## To connect shapes using the standard connector:

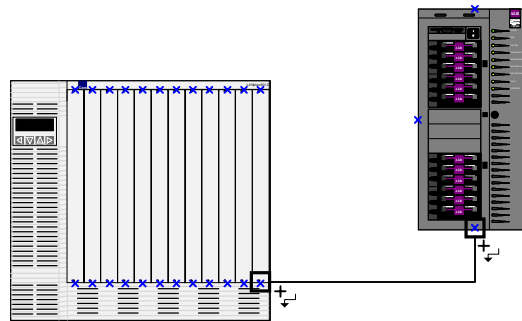


- 1 Choose the connector tool.
- 2 Point to a connection point on one of the shapes, then click the mouse button and drag the pointer to a connection point on the other shape.

The shapes are connected. Their connection points turn red, indicating that they are glued.

**Figure 3-4**

Connecting shapes using the connector tool



...then click the mouse button and drag the pointer to a connection point on the other shape.

With the connector tool chosen, point to a connection point on one shape...

## To connect shapes using other connector types:

- 1 Choose Tools > Macro > VNE > Locate Network Equipment to open the Locate Network Equipment window.
- 2 In the Locate Network Equipment window, click the Tools Logical Shapes group, then click Connectors And Callouts.
- 3 Drag the desired connector out of the stencil.
- 4 Point to a connection point (x) on one of the shapes.
- 5 Click the mouse button and drag the pointer to a connection point on the other shape to connect the shapes.

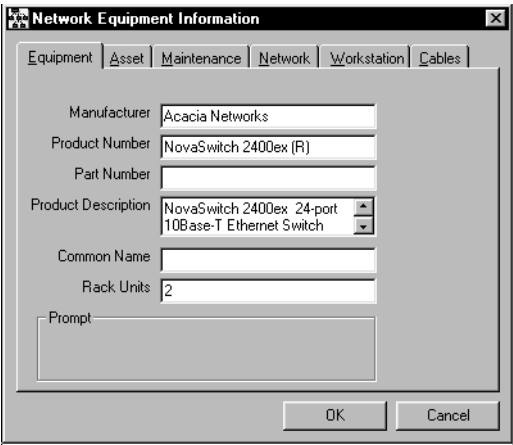
# Creating reports

Each VNE shape has properties associated with it. For all manufacturer shapes, the manufacturer, product name, product description, and model number are prefilled. You can assign additional properties to shapes, and then use these properties to track equipment and generate reports, such as an inventory report, a cost sheet, or a bill of materials.

**To view or change the properties of a shape:**

- 1 Right-click a shape, then choose Properties from the shortcut menu.
- 2 Click the tab for the property category you want to view or change. Following are descriptions of the tabs and the property categories each contains.

**Figure 3-5**  
The Properties dialog box has tabs for commonly used categories of network equipment data.



**Equipment** Manufacturer, Product Number, Part Number, Product Description, Common Name, and Rack Units (which specify the amount of space that 1-D shapes take up on a rack)

**Asset** Asset Number, Serial Number, Location, Building, Room, Department, and Employee

**Maintenance** Purchase Date, Cost, Warranty Exp., Installed Date, Support Agreement, Comments, Support Contact Name, and Contact Phone

**Network** Network Name, IP Address, Subnet Mask, Administrative Interface, and Number Of Ports

**Workstation** Operating System, Memory, Hard Drive Size, Quantity, Additional Devices, and NIC ID#

**Cables** Cable Type, Cable Length, Cable Source, Cable Destination, and Drop Number

- 3 Click OK.

**To create reports:**

- 1 Choose Tools > Property Report.
- 2 Follow the instructions in the Property Reporting Wizard.

Your report can include all shapes, or only the ones you select. You can select by layer to report on specific manufacturers or equipment types.

## Modifying a network shape and using the VNE Property Duplicator

You can use one shape as the basis for a new, adapted shape. For example, if a new model of a server looks very much like the previous one, you can create a copy with a different model name or number.

If you want to use the modified shape only in the current drawing, you can just drag the appropriate master out of its stencil and edit it. If you want to make the new shape reusable, you can drag it into a stencil to create a new master.

You can add shapes to a stencil only when an original stencil is open. We recommend placing your modified shapes in a new, blank stencil reserved for your own custom shapes.

You can also create a shortcut to your newly modified shape in the Locate Network Equipment utility so that you can use it to locate the shape later.

The VNE Property Duplicator restores custom properties and connections to a shape you have modified.

**To modify a shape:**

- 1 Drag the shape out of the stencil and drop it onto the drawing page.
- 2 Ungroup the shape so that you can edit it: Select the shape, then press Ctrl+U.

A message box tells you that ungrouping the shape severs its link to the master. If the message box doesn't appear, it means the shape has several grouping levels. Continue to ungroup by pressing Ctrl+U until the message box appears.

- 3 Click OK in the message box.

The shape is separated into its parts, and the connection with the master is broken.

- 4 Make the desired modifications.

For example, to change the model name on the faceplate, delete the current text, then type the new text, using the same font and size.

- 5 When you are done modifying the shape, select all of its component shapes, then press Ctrl+G to create a group.
- 6 Convert the new shape to Windows Metafile format: With the shape still selected, press Ctrl+X to cut the shape, then choose Edit > Paste Special.
- 7 In the Paste Special dialog box, select Picture (Enhanced Metafile), then click OK.

A copy of the shape in Windows Metafile format is pasted into the drawing.

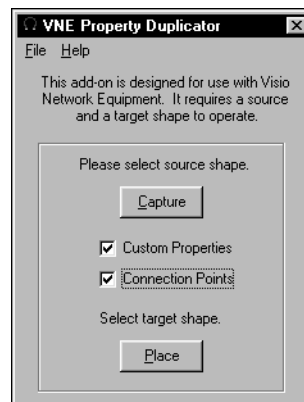
Ungrouping a shape for editing severs its link to its master, so that the edited shape loses its connection points and custom properties. To restore custom properties and connections to a shape you have modified, use the VNE Property Duplicator.

**To add properties to a modified shape:**

- 1 Select a shape on the drawing page (the “source” shape) whose custom properties or connection points you want to copy.
- 2 Choose Tools > Macro > VNE > VNE Property Duplicator.  
**TIP** If necessary, drag the VNE Property Duplicator window so that you can see both the source and target shapes.
- 3 Click Capture.
- 4 Select the options you want to transfer: Custom Properties, Connection Points, or both.
- 5 Select your modified shape (the “target”), then click Place.  
The selected attributes are copied to the target shape. Click OK.
- 6 *Optional* Delete the source shape from the diagram.

Repeat the steps to edit more shapes, or close the VNE Property Duplicator.

**Figure 3-6**  
The VNE Property Duplicator



**To save a modified shape on a stencil:**

- 1 Choose File > Stencils > Blank Stencil.
- 2 Press Ctrl+drag the shape onto the blank stencil.

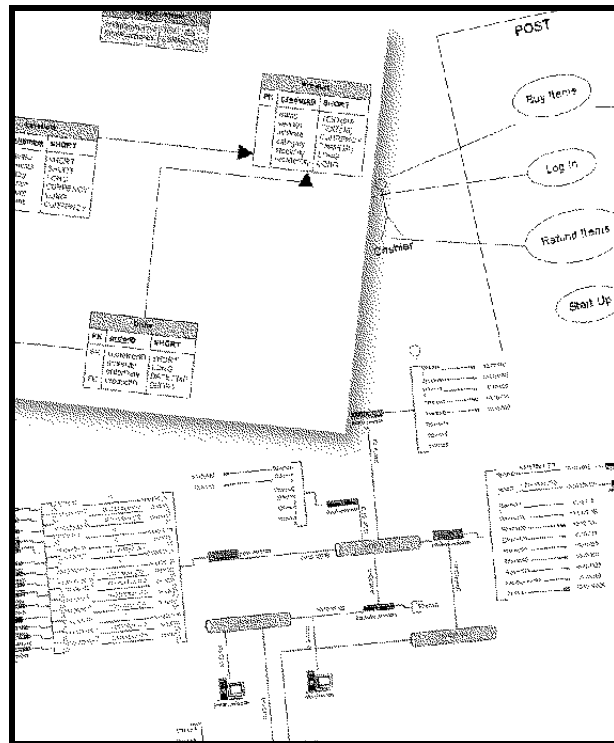
A copy of the shape is added to the stencil with a default master name. The stencil window is now active, and the menu bar changes to include stencil commands.
- 3 To rename the shape, with the stencil window still active, choose Master > Properties. In the Master Name section, replace the default name with the one you want.
- 4 Choose File > Save so you can name the stencil and save it.

**To create a shortcut to your modified shape in the Locate Network Equipment utility:**

- 1 In the Locate Network Equipment utility, click the Open Documents group in the shortcut bar.
- 2 Click the folder for your custom stencil (created in the previous procedure) to display its contents in the results window.
- 3 In the shortcut bar, click the group in which you want to create the shortcut. For example, click General.
- 4 Choose File > New Folder.
- 5 Drag the modified shape from the results window into the new folder.

The modified shape is added to the folder as a shortcut and is saved when you close the Locate Network Equipment utility.





## Modeling Software with Visio Enterprise



# Modeling software components

Visio® Enterprise supports all the major object-oriented and data-oriented software modeling languages, including the Unified Modeling Language (UML), recently adopted as a standard by the Object Management Group, Inc. In addition, Visio Enterprise provides the tools you need to prototype and design user interfaces.

This chapter lists the range of software modeling solutions available in Visio Enterprise and explores in detail the Visio Enterprise approach to the UML.

## Topics in this chapter

- Modeling software components in Visio Enterprise ..... 60
- Software modeling solutions ..... 61
- The Unified Modeling Language in Visio Enterprise ..... 63

# Modeling software components in Visio Enterprise

In support of software modeling and interface design, Visio Enterprise offers 19 solutions. Using the solutions, you can

- **Create integrated system models using the UML** that you can export to Microsoft Repository, where they can be accessed by other developers.
- **Reverse engineer information models** from Microsoft Repository or from source code created in Microsoft Visual C++ 5.0 and 6.0 or created in Microsoft Visual Basic 5.0 and 6.0 to generate UML static structure models.
- **Design and document software projects** using object-oriented methodologies, such as Booch, ROOM, and Rumbaugh; or data-modeling methodologies, such as Chen and Martin.
- **Show the procedural components of programs** using structured charts, such as Nassi-Schneiderman charts.
- **Mock up Microsoft Windows 95 and Office 97 UI components**, such as dialog boxes, menus, and wizard screens, so you can usability test and revise them before including them in a product.

Each software modeling solution consists of SmartShapes® symbols programmed to behave in accordance with the rules of the specific modeling methodology. In addition, the UML solution builds models that it displays as hierarchical tree views in the UML Navigator and includes a semantic error checker that runs in the background to identify errors.

For details about the UML, see [“The Unified Modeling Language in Visio Enterprise”](#) later in this chapter, or [Chapter 5, “Creating software models with the UML.”](#)

# Software modeling solutions

This section briefly describes each Visio Enterprise software modeling and interface design solution. You can open these solutions by choosing File > New > Software Diagram or File > New > UML, and then choosing the solution you want to use.

Each software modeling and UI design solution includes context-sensitive online help. The table that follows the list of solutions describes how you can access this help.

**Booch OOD** Supports the creation of object, class, timing, state, module, and process diagrams using Booch object-oriented design notation.

**Chen ERD** Supports the creation of entity relationship and data structure diagrams using Chen ERD notation.

**COM And OLE** Supports the creation of COM (Component Object Model) objects and interfaces and system diagrams.

**Express-G** Supports the creation of entity-level and schema-level diagrams using Express-G notation, the graphical component of the Express formal information requirements specification language.

**Fusion** Supports the creation of diagram types included in the Fusion methodology.

**Gane-Sarson DFD** Supports the creation of data flow diagrams using Gane-Sarson notation.

**Jackson** Supports the creation of data and program structures using the Jackson software design method.

**Jacobson Use Cases** Supports the creation of use case, state transition, interaction, and other diagrams included in the Jacobson object-oriented software engineering method.

**Martin ERD** Supports the creation of object-oriented analysis and design diagrams using Martin ERD notation.

**Nassi-Schneiderman** Supports the representation of sequence, selection, case, and repetition in program structure charts.

**Program Structure** Supports the creation of structural diagrams of programs and memory objects.

**ROOM** Supports the creation of Real-Time Object-Oriented Modeling (ROOM) diagrams that show the relationships among system components and system responses to events.

**Rumbaugh OMT** Supports the creation of object, dynamic, and functional diagrams using the Rumbaugh Object Modeling Technique.

**Shlaer-Mellor OOA** Supports the creation of class diagrams, class structure charts, dependency diagrams, and inheritance diagrams using Shlaer-Mellor notation.

**SSADM** Supports the creation of logical data structures, data flow diagrams, entity life histories, and other system analysis and design diagrams.

**System Structure** Supports the creation of diagrams that represent Windows and Macintosh operating systems and user interfaces (for Windows 95 or Office 97 interfaces, see the Windows User Interface solution).

**UML** Supports the creation of an integrated system model based on the Unified Modeling Language diagram types. From the UML solution, you can export static structure models to Microsoft Repository 2.0, and you can reverse engineer Repository UML models or source code to create static structure models in Visio Enterprise.

**Windows User Interface** Supports users who need to develop, document, or prototype Office 97 and Windows 95 user interfaces.

**Yourdon And Coad** Supports the creation of object state, data flow, and other diagrams included in Yourdon and Coad object-oriented analysis and design notation.

## Online help for software modeling and user interface design solutions

<i>Information source</i>	<i>What it is</i>	<i>How to find it</i>
Shape-specific help	A window of information that explains how to work with a shape.	Right-click any shape, then choose Shape Help from the shortcut menu.
Template-specific help	Help that describes how to create specific diagrams, including the most efficient sequence in which to draw, how shapes work together, and tips and tricks.	Choose Help > Template Help, double-click Visio Templates, double-click Software Diagram or UML, then double-click the type of solution you want.
Visio help	Conceptual and procedural help related to working in Visio Enterprise and in specific solutions. Also defines specialized terms.	Choose Help > Visio Help. To search for a term, click the Index tab, then type the term you want.
Dialog box option help	Help about what each option in a dialog box is used for and what information to choose or type to achieve a specific result.	Click the Help button in the lower right corner of a dialog box.
Screen tips for toolbar buttons and control handles	Pop-up tips that tell you what toolbar buttons can be used for or how to use a control handle on a shape.	Float the pointer over the toolbar button or control handle to see a tip.

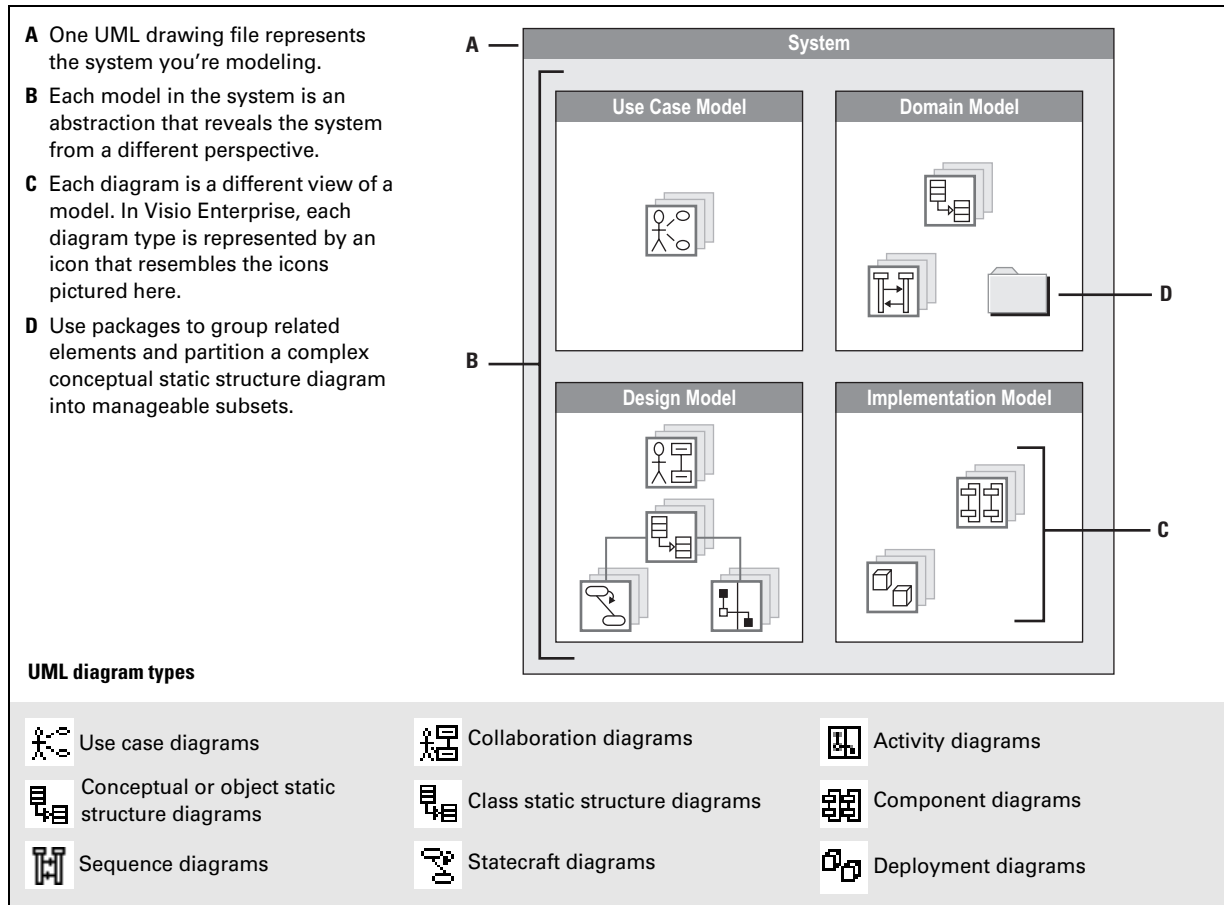
## The Unified Modeling Language in Visio Enterprise

The Unified Modeling Language is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system. The UML is a derivative language, formed from parts of three earlier languages: Booch, OMT (Object Modeling Technique), and OOSE (Object-Oriented Software Engineering). The Object Management Group, an international organization with more than 800 members, has adopted the UML as a standard modeling language.

The vocabulary of the UML is a **notation**—a set of shapes in which each shape has a particular meaning. Its grammar consists of a carefully defined **semantics** that describes how each shape can be used. In combination, the notation and semantics make it possible to model all kinds of systems, regardless of their scope and complexity. The authors of the *UML Semantics Guide* compare the UML to a blueprint for a construction project: It helps a team visualize a system's architecture throughout the development cycle.

The UML notation includes elements that support the creation of several diagram types. Each diagram provides a different view of one model of a software system.

**Figure 4-1** Modeling a software system using the UML



Understanding the Visio Enterprise approach to the UML will help you work more efficiently. The annotated illustrations in the following sections present the Visio Enterprise UML shapes and interface, and describe the Visio Enterprise approach to creating models in the UML.

You can also consult the complete *OMG Unified Modeling Language Specification*, which is included as a PDF file with Visio Enterprise. During installation, choose to install the Visio Documentation component, and then look for the file in the \Visio\Docs folder. To read a PDF file, you must install Adobe Acrobat Reader: on the Visio Enterprise CD, in the Acrobat folder, double-click the Ar32e301.exe file.

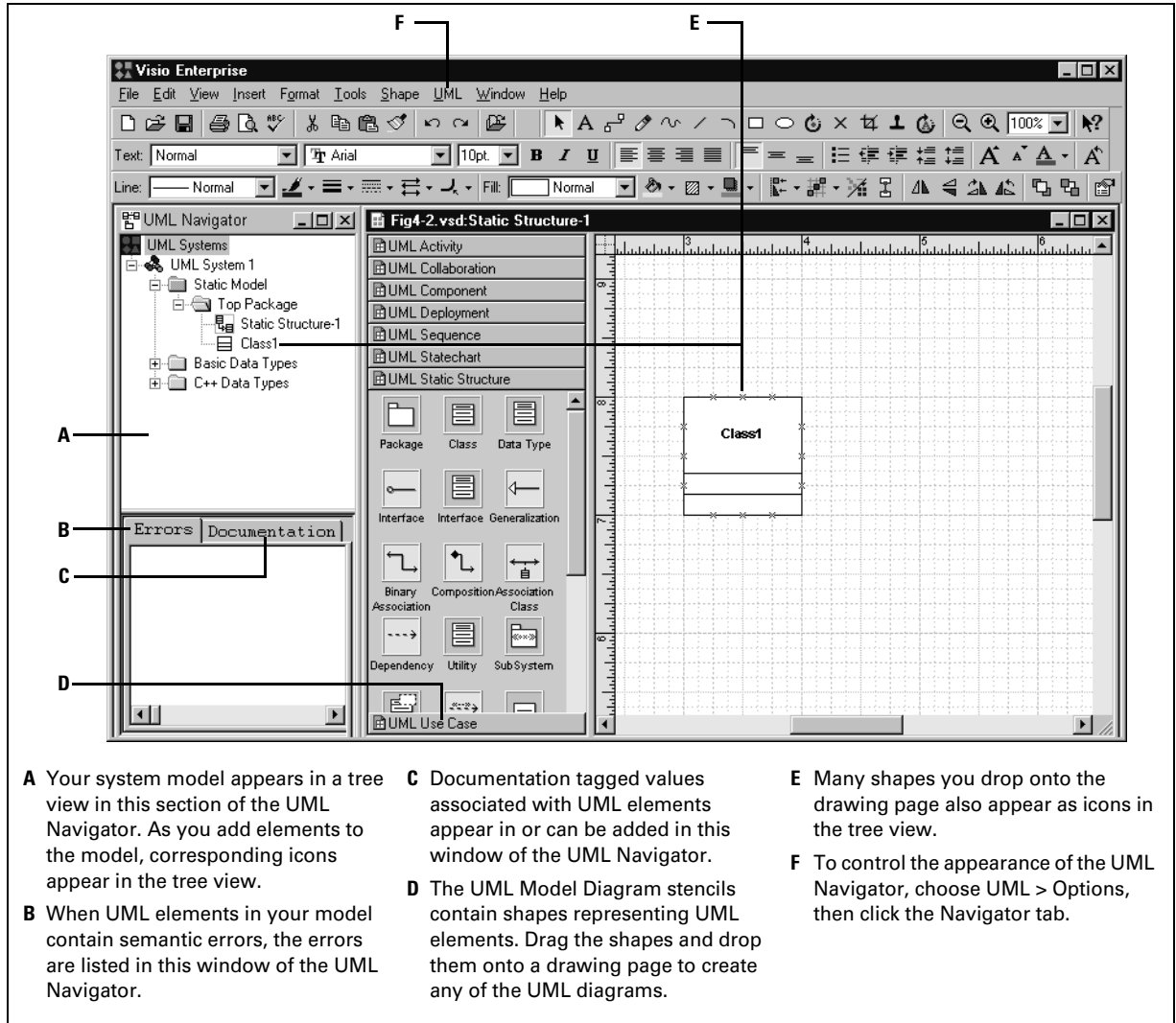


## One UML drawing file represents a system

In Visio Enterprise, the model you develop for a **system**, such as a new computer game, a library information system, or a process for assembling microchips, is contained within one UML drawing file. When you save your system model, Visio Enterprise gives the file a .vsd extension.

To open the UML solution, choose File > New > UML > UML Model Diagram.

**Figure 4-2** The UML drawing file



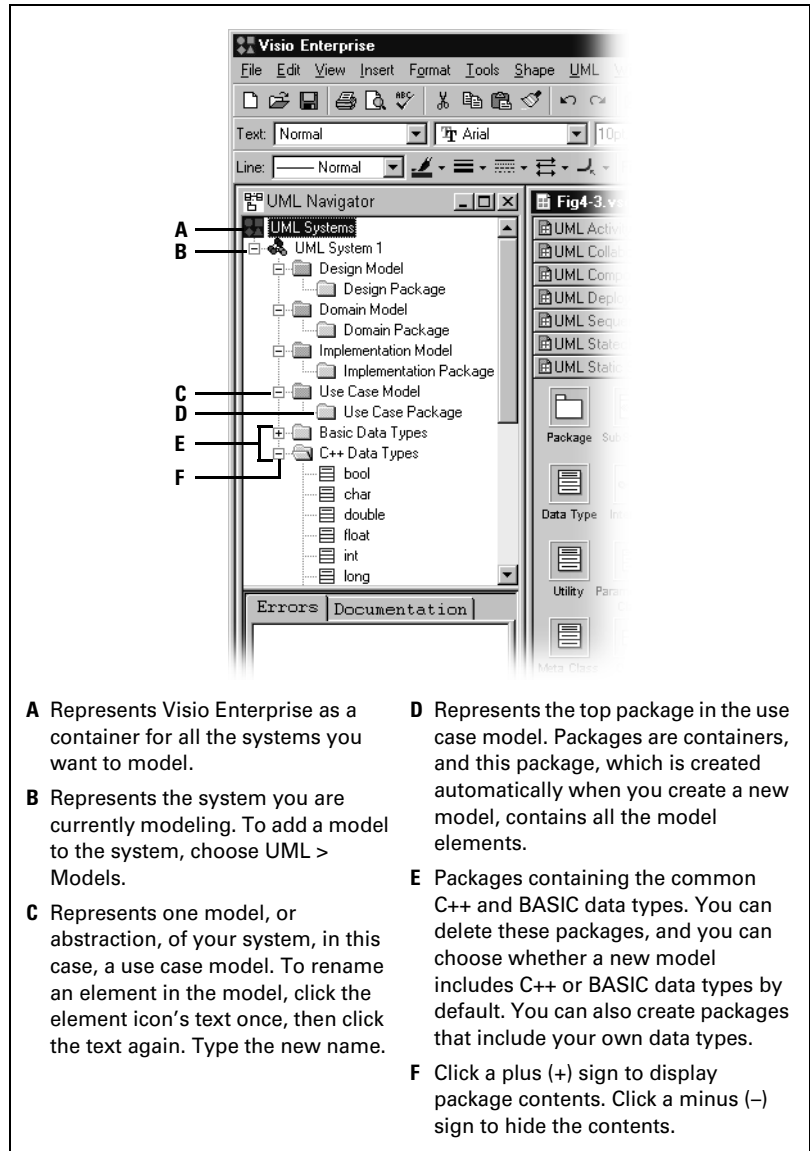
## One system includes many models

To decompose a complex system into understandable chunks, you can create different **models**, or abstractions, of the system. In Visio Enterprise, all the models appear together in the same drawing file, and each model is visible in the tree view.

To open the UML solution, choose File > New > UML > UML Model Diagram.

**Figure 4-3**

A model usually relates to a specific phase of development.



- A** Represents Visio Enterprise as a container for all the systems you want to model.
- B** Represents the system you are currently modeling. To add a model to the system, choose UML > Models.
- C** Represents one model, or abstraction, of your system, in this case, a use case model. To rename an element in the model, click the element icon's text once, then click the text again. Type the new name.
- D** Represents the top package in the use case model. Packages are containers, and this package, which is created automatically when you create a new model, contains all the model elements.
- E** Packages containing the common C++ and BASIC data types. You can delete these packages, and you can choose whether a new model includes C++ or BASIC data types by default. You can also create packages that include your own data types.
- F** Click a plus (+) sign to display package contents. Click a minus (-) sign to hide the contents.

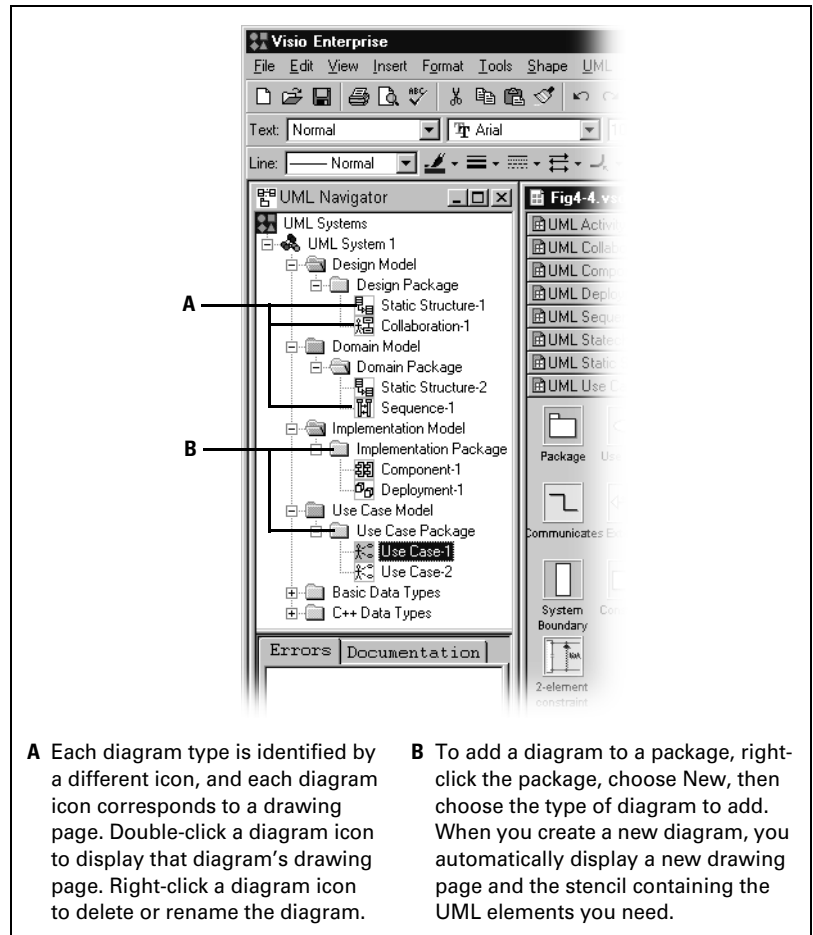
## View one model in different ways

To thoroughly consider a system, you must approach it from different perspectives or **views**. In the UML, a view corresponds to a diagram. For example, use case diagrams provide a look at the system from a user's point of view. Class static structure diagrams take user requirements and translate them into software classes and relationships.

To open the UML solution, choose File > New > UML > UML Model Diagram.

**Figure 4-4**

Each UML diagram presents one view of a model.

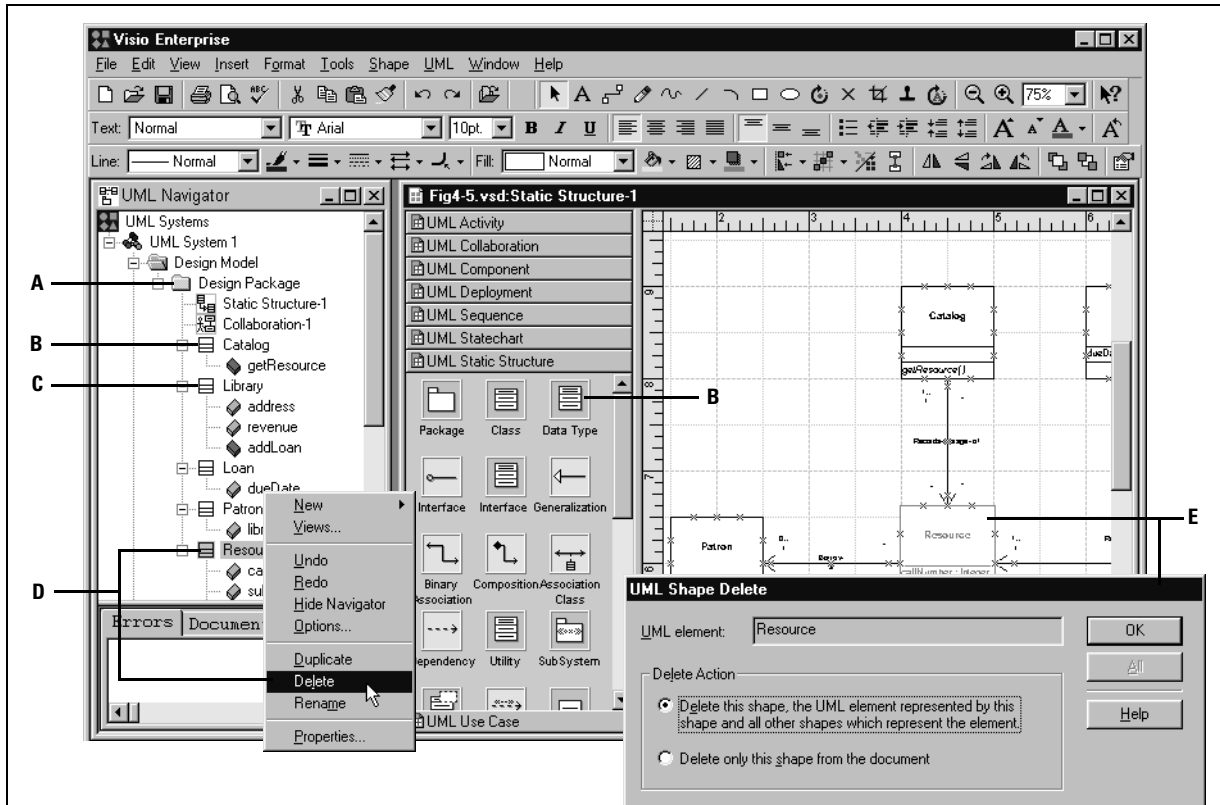


## Comparing the tree view and the diagrams

The tree view provides a picture of the models of the system, and is a comprehensive view of most elements in a system's models. Each diagram, which is composed of shapes representing UML elements, is just one view of one model in the system, a snapshot of a model. Each shape is a view of one UML element.

To open the UML solution, choose File > New > UML > UML Model Diagram.

**Figure 4-5** Tree view and diagram



**A** Packages are a model's containers. To add an element to a package, right-click the package icon, choose New, then choose the element you want. Under a package, first diagrams and then elements are listed alphabetically by name.

**B** To add one view of an element to a diagram, drag either a shape from a stencil or an icon from the tree view and drop it onto the drawing page.

**C** Elements may appear in the model that are not represented graphically in any diagram.

**D** If you delete an element from the tree view, Visio Enterprise removes the element and all of its views (shapes) from the diagrams.

**E** By default, if you delete a shape from a drawing page, Visio Enterprise gives you a choice: You can remove the element from both the model and the diagram or just from the diagram. You can customize the default delete behavior.

## Using packages

A **package** is the basic organizing element of a UML model. You can think of the entire system as a package that contains all the other packages, diagrams, and elements. An element is owned by only one package, but you can include a reference to the element in other packages.

To open the UML solution, choose **File > New > UML > UML Model Diagram**.

**Figure 4-6** Packages and elements

The screenshot shows the Visio Enterprise interface. On the left is the UML Navigator tree view. In the center is a stencil with various UML shapes. On the right is a drawing page titled 'Fig4-6.vsd:Static Structure-2' containing a UML class diagram. Three callouts labeled A, B, and C point to specific actions in the interface.

**A** To add a new element to a package in the model, right-click the package icon (resembles a manila folder) in the tree view, choose **New**, then choose the element you want to add.

**B** If you drag an icon for an element in one package and drop it into a diagram in another package, Visio Enterprise creates a view of the element in the second package. The element's name includes a reference to its home package. For example, the **Library** and **Catalog** classes in this drawing originated in the **Design** package.

**C** When you add an element to a diagram by dragging a shape from a stencil onto the drawing page, the element is also added to the package in the model that contains the diagram (or view). For some elements, an icon appears in the tree view.

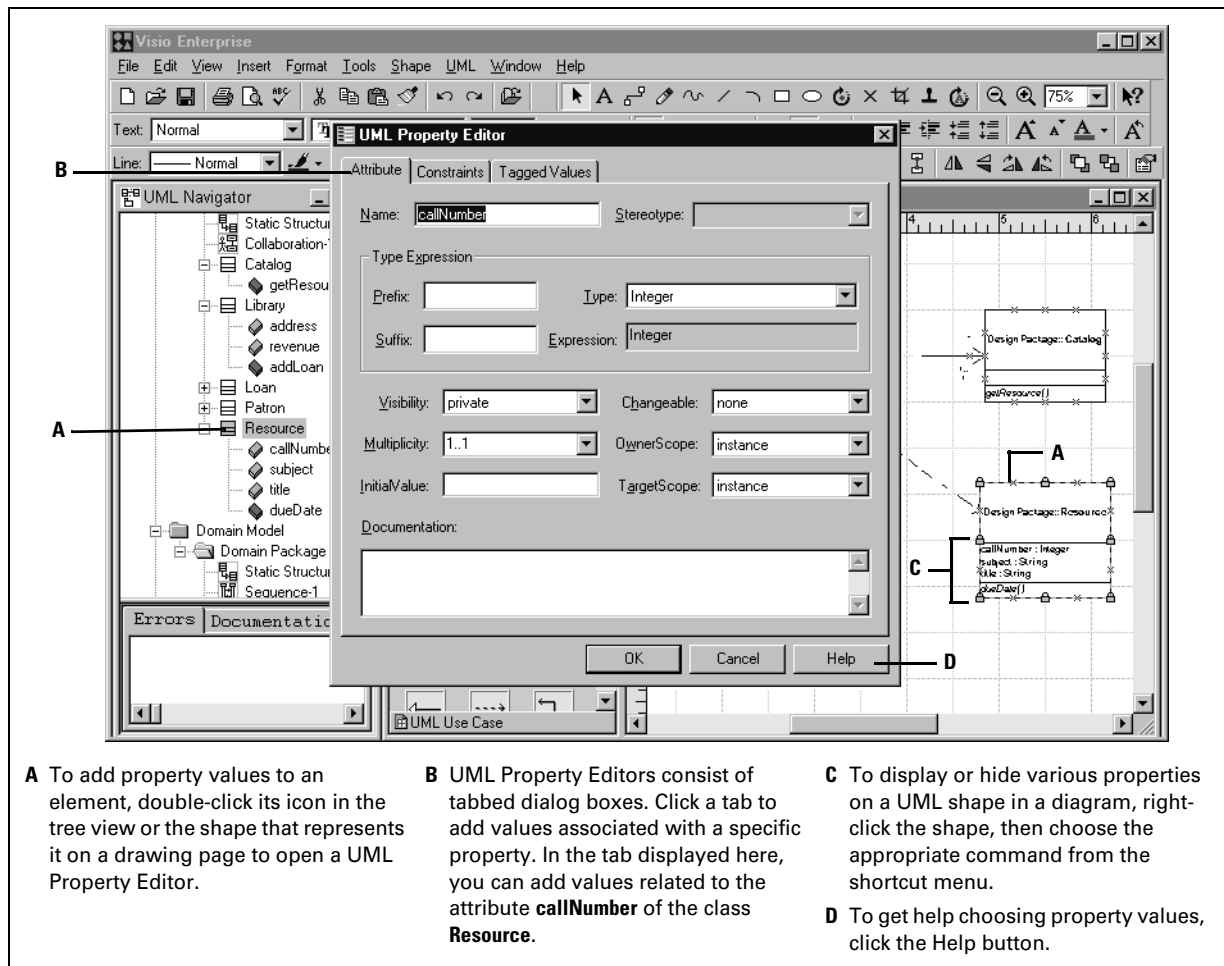
## Adding property values to UML elements

Properties are values you associate with UML model elements that become integral parts of those elements. For example, you can associate property values, such as attributes and operations, with classes in class static structure diagrams. You can also add discretionary values called tagged values to any UML element. In Visio Enterprise, you add property values to elements using the elements' easily accessible UML Property Editors.

When you export a model to Microsoft Repository, all the property values get exported along with the model elements.

To open the UML solution, choose File > New > UML > UML Model Diagram.

**Figure 4-7** UML element properties



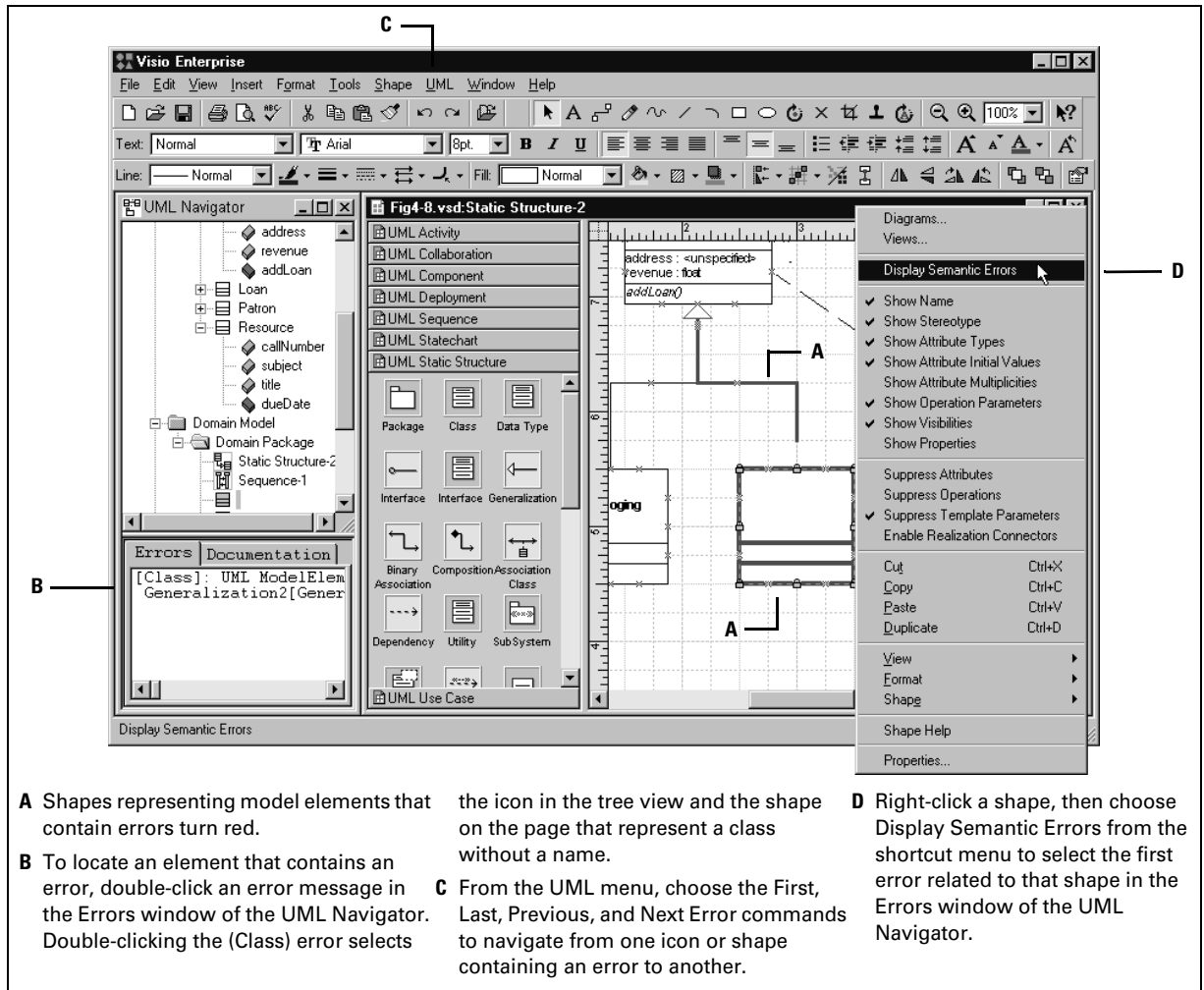
## Conforming to the UML syntax

As you model a system using the UML, Visio Enterprise runs a **semantic error checker** in the background that identifies and diagnoses errors, such as missing data and improper use of the notation. In addition to diagnosing errors, the semantic error checking feature in the UML solution helps you locate the shapes or icons that represent elements that contain errors.

For details about semantic error checking, see “[Diagnosing and fixing semantic errors](#)” in [Chapter 5, “Creating software models with the UML.”](#)

To open the UML solution, choose **File > New > UML > UML Model Diagram**.

**Figure 4-8** Semantic error checking

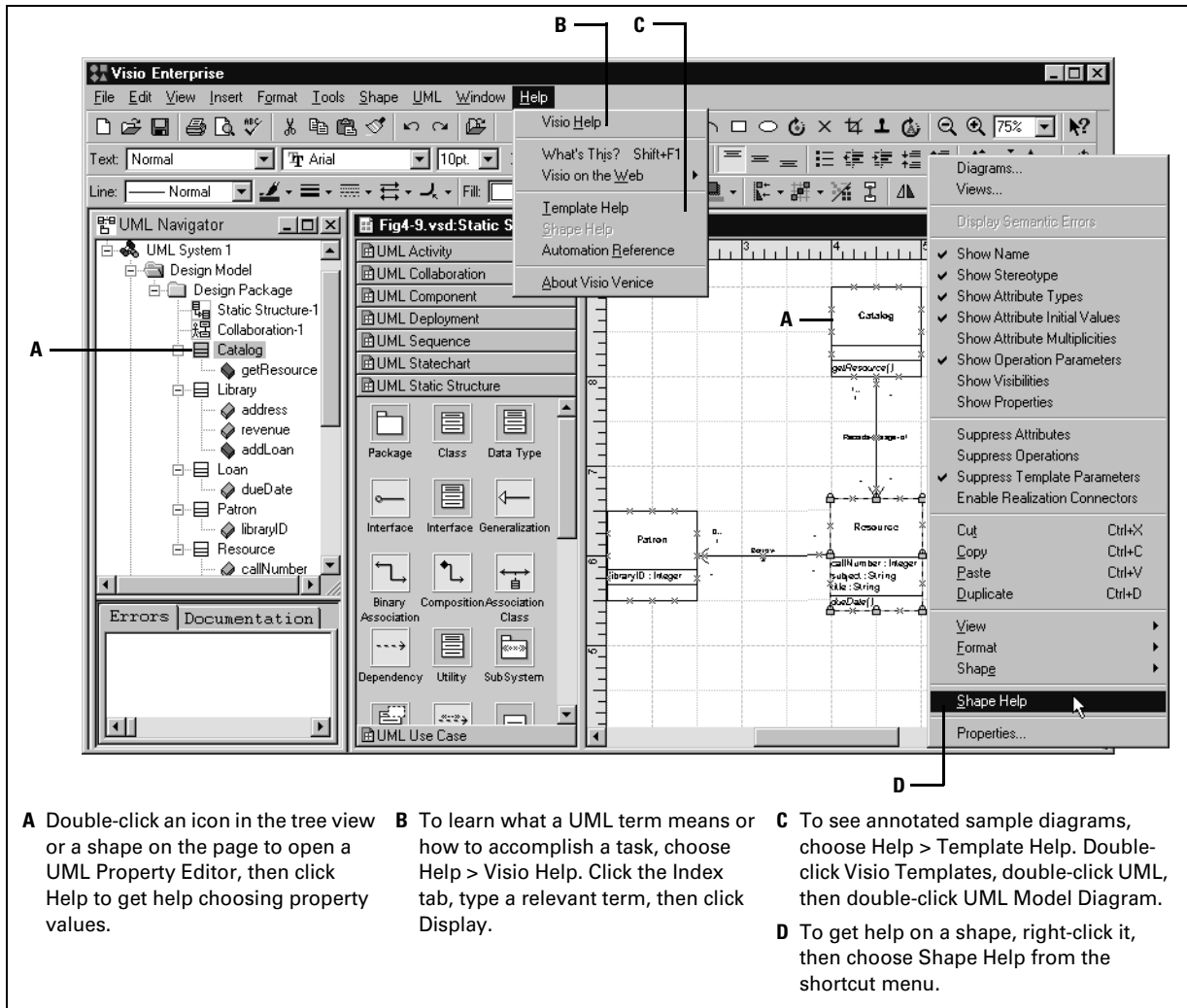


## Getting help

The UML solution in Visio Enterprise is supported by a system of easily accessible online help. In online help, you can learn the meaning of a UML term, how to work with a specific diagram type or shape, what values to use for a specific property, and more.

To open the UML solution, choose File > New > UML > UML Model Diagram.

**Figure 4-9** Online help for the UML solution





# Creating software models with the UML

The UML solution in Visio® Enterprise offers full support for the collaborative creation of object-oriented software system models. In this chapter, you can learn how to build a model from scratch, or how to import a model from Microsoft Repository or generate one from Visual C++ or Visual Basic source code. You can also learn how to export models you create to Microsoft Repository, where they can be accessed by other developers or imported into CASE tools to generate source code. Because you can share models with other developers, Visio Enterprise is an excellent tool for team-based development.

## Topics in this chapter

- How to model a system using the UML solution..... 74
- The UML diagrams ..... 84
- Storing and retrieving models: working with Microsoft Repository ..... 99
- Reverse engineering source code ..... 102

# How to model a system using the UML solution

The Visio Enterprise UML solution offers full support for modeling a complex software system. The solution includes the following tools, shapes, and functionality:

- **The UML Navigator**, which provides a tree view view of your system model and a means of navigating from one view to another.
- **Predefined shapes** representing all the elements in the UML notation that support the creation of all UML diagram types. The shapes are programmed to behave in ways that are consistent with the UML semantics.
- **Easy-to-access UML Property Editors**, where you can add attributes, operations, and other properties to UML elements.
- **Semantic error checking**, which operates dynamically to identify and diagnose errors, such as missing data and improper use of the notation.
- **Support for Microsoft Repository 2.0**, which allows you to share static structure models with other developers by importing them to or exporting them from Microsoft Repository.
- **The ability to reverse engineer Visual C++ and Visual Basic source code** created in Microsoft Visual C++ or Microsoft Visual Basic.
- **Context-sensitive online help**, which you can use to find the meanings of UML-specific terms, learn how to work with shapes and create specific diagram types, choose property values, and more.

You can also consult the complete *OMG Unified Modeling Language Specification*, which is available in PDF format with Visio Enterprise. During installation, choose to install the Visio Documentation component, and then look for the file in the \Visio\Docs folder. To read a PDF file, you must install Adobe Acrobat Reader: on the Visio Enterprise CD, in the Acrobat folder, double-click the Ar32e301.exe file.

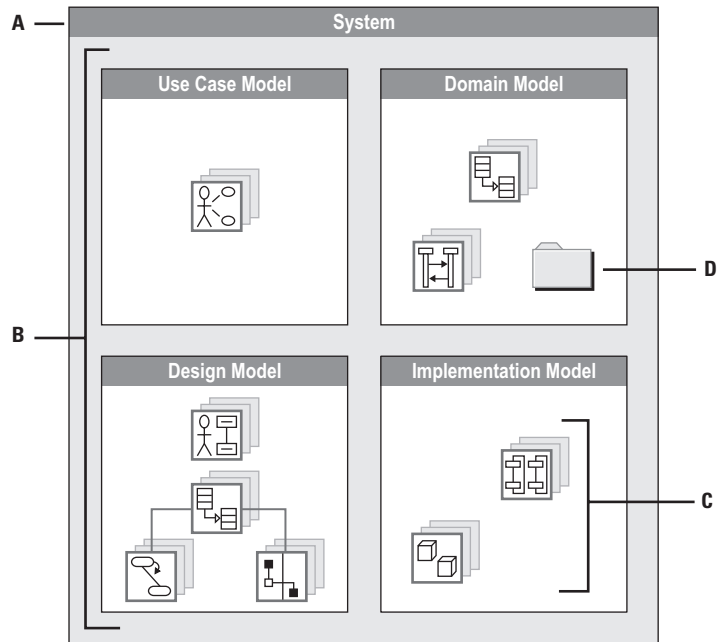
For details about the UML environment, see [“The Unified Modeling Language in Visio Enterprise”](#) in Chapter 4, “Modeling software components.”

## Modeling a system using the UML

Visio Enterprise supports the creation of complex system models and all of the UML diagram types. Figure 5-1 shows a conceptual overview of how you might create models of a system in Visio Enterprise.

**Figure 5-1** Creating a system model in Visio Enterprise

- A** One UML drawing file represents the system you're modeling.
- B** Each model in the system is an abstraction that reveals the system from a different perspective.
- C** Each diagram is a different view of a model. In Visio Enterprise, each diagram type is represented by an icon that resembles the icons pictured here.
- D** Use packages to group related elements and partition a complex conceptual static structure diagram into manageable subsets.



### UML diagram types



Use case diagrams describe the external actors and how they use the system.



Conceptual models (object diagrams), a type of static structure diagram, decompose a system into related real-world objects.



System sequence diagrams show, for a particular use case, the system events that external actors generate.



Collaboration diagrams show how objects involved in each system event interact via messages to fulfill tasks.



Class diagrams, a type of static structure diagram, translate real-world objects into software entities with associations, attributes, interfaces, methods, and dependencies.



For state-dependent objects, statechart diagrams describe how objects react to different events.



Activity diagrams show the workflow related to individual objects.



Component diagrams show the structure of the code itself and the dependencies between components.



Deployment diagrams show the physical relationships among the software and hardware components.

To begin modeling a system, open the Visio Enterprise UML Model Diagram template, and then perform these iterative steps:

- Add a new model to the existing system.
- Start a new diagram within a model.
- Add UML elements to a model.
- Associate properties with UML elements.
- Diagnose and fix semantic errors.

The following sections walk you through each step. For details about the UML diagram types, see [“The UML diagrams”](#) later in this chapter.

## Modeling a new system

To begin modeling a software system, open the Visio Enterprise UML Model Diagram template. By default, a new system includes the data types commonly used in C++. You can also access the data types commonly used in BASIC, and you can delete data types you don’t use and add data types of your own.

### To begin modeling a system:

- In Visio Enterprise, choose File > New > UML > UML Model Diagram.

The modeling environment you open includes the UML Navigator, a group of stencils, and a drawing page. The top window of the UML Navigator includes icons arranged in a tree view. The icons represent the UML model elements.

**TIP** If you frequently start new system models, you can create a shortcut to the UML Model Diagram template and place the shortcut on your desktop. To create a shortcut, right-click UML Model Diagram.vst located in the \Visio\Solutions\UML folder, then choose Create Shortcut. Drag the shortcut icon onto your desktop.

### To delete the existing C++ or BASIC data types:

- In the tree view, right-click the icon representing the package containing the C++ or BASIC data types you want to delete, then choose Delete from the shortcut menu.

The package and all the data types it contains disappear from the tree view. To undo this deletion, choose UML > Undo.

**TIP** You can control the data types that are included by default in a new system model. To choose the default data types, choose UML > Options, then click the UML Document tab. Uncheck the data types you don’t want to include.

**To add data types to a system:**

- 1** Right-click the UML System icon in the tree view, choose Packages, then, in the UML Packages dialog box, click New.
- 2** In the UML Property Editor, type a name for the data types package (for example, *My Data Types*), enter or choose other package properties, then click OK. Click OK again.

Visio Enterprise adds a new package to the tree view named My Data Types.

- 3** Right-click the new package, then choose New > Datatype from the shortcut menu.
- 4** Type a name for the data type, enter or choose the other properties you want, then click OK.

Visio Enterprise adds an icon for the new data type to the tree view inside the My Data Types package.

- 5** Repeat steps 3 and 4 until you have added all the data types you need.

**TIP** You can also add interfaces to a system in the same way you add data types. Follow steps 1 and 2 above to create a new interface system-level package. Right-click the package, then choose New > Interface to add a new interface to the package.

For details about models and the tree view, see [“The Unified Modeling Language in Visio Enterprise”](#) in [Chapter 4, “Modeling software components.”](#)

## Adding a new model to an existing system

When you open the UML Model Diagram template, your system already includes one model called Static Model. You can edit the model by changing its name and setting its properties. You can also add new models that offer different perspectives of your system, and delete models you no longer need. A system must include at least one model, so Visio Enterprise will prevent you from deleting the last available model.

### To edit an existing model:

- 1 In Visio Enterprise, open the UML Model Diagram file that represents your system.
- 2 Choose UML > Models.
- 3 In the UML Models dialog box, select the model to edit, then click Edit.
- 4 In the UML Property Editor, enter or choose values for the properties you want to change. Click OK, then click OK again.

**TIP** You can also edit model properties by double-clicking the icon that represents the model. To change just a model's name, click the name in the tree view once, then click the name again. Type the new name.

### To add a new model:

- 1 Open the UML Model Diagram file that represents your system.
- 2 Choose UML > Models.
- 3 In the UML Models dialog box, click New.
- 4 In the UML Property Editor, enter or choose values for the properties you want. Click OK, then click OK.

### To delete a model:

- 1 Open the UML Model Diagram file that represents your system.
- 2 Choose UML > Models.
- 3 In the UML Models dialog box, select the model to delete, click Delete, then click OK.

**TIP** You can also delete a model by right-clicking its icon in the tree view, then choosing Delete.

## Starting a new diagram within a model

Visio Enterprise is designed to help you create UML diagrams in the proper relationship to one another. For example, you can add a new static structure or use case diagram to any package, but you can create a statechart or activity diagram only in relation to a specific class or use case.

**NOTE** To open the UML solution, choose File > New > UML > UML Model Diagram.

### To start a new diagram:

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, right-click the icon representing the package, class, or use case that you want to contain the new diagram.
- 3 Choose New from the shortcut menu, then choose the type of diagram you want to create.

Visio Enterprise adds an icon representing the diagram to the tree view. It also automatically displays the stencil containing the shapes appropriate for the new diagram you are creating and a blank drawing page.

**TIP** You can also create a new diagram by right-clicking a package, class, or use case shape on the drawing page, then choosing Diagrams from the shortcut menu. In the UML Diagrams dialog box, click New, then choose the type of diagram you want.

For details about each diagram type, see [“The UML diagrams”](#) later in this chapter.

## Adding UML elements to a model

A diagram is a selective view of a model. Elements, such as classes, use cases, and components, can exist in a model but never appear in a diagram. In Visio Enterprise, this relationship is handled in the following ways:

- The tree view in the UML Navigator represents the system. You create diagrams by dropping shapes or tree view icons onto a drawing page.
- When you add an element to a diagram by dragging a shape from a stencil and dropping it onto the drawing page, the element is automatically added to the tree view.
- When you add an element to the tree view, the element does *not* appear in a drawing unless you drag the icon from the tree view and drop it onto a drawing page.
- You can duplicate icons in the tree view, and you can drag element icons into different packages.

- You can add different views of the same element to different drawings by dragging the icon representing the element onto different drawing pages.

For more details about the drawing page versus the tree view, see “Comparing the tree view and the diagrams” in Chapter 4, “Modeling software components.”

#### **To add a new element to a model:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, right-click the package or class in which you want to create a new element, then choose New from the shortcut menu.
- 3 Choose the kind of element you want to create.

Visio Enterprise adds an icon for the new element to the tree view.

**TIP** To delete an element from the tree view, right-click the icon that represents the element, then choose Delete.

#### **To move an element in the tree view:**

- Drag the element into a new package or other container.

#### **To duplicate an existing element in the tree view:**

- 1 Right-click the element you want to duplicate, then choose Duplicate from the shortcut menu.

Two elements in the same package cannot have the same name, so when you duplicate an element, Visio Enterprise automatically gives the duplicate a new name.

- 2 Drag the duplicate icon to the appropriate package in the tree view.

**TIP** You can also duplicate an item by pressing Ctrl+dragging the icon to a new package.

#### **To add a new element to a diagram:**

- 1 In the tree view, double-click the icon that represents the diagram to which you want to add a new element.

The appropriate stencil and drawing page appear.

- 2 Drag the shape representing the element you want to add to a diagram from the stencil and drop it onto the drawing page.
- 3 When you drop classifier shapes, such as Class, Use Case, and Component shapes, onto the drawing page, Visio Enterprise automatically adds an icon for each new element to the tree view.

**TIP** You can also add an element to a diagram by creating a new element in the tree view, then dragging the element onto the drawing page.



**To add a view of an existing element to a diagram:**

- 1 In the tree view, double-click the icon that represents the diagram to which you want to add a view of the element.

The appropriate stencil and drawing page appear.

- 2 In the tree view, locate the icon of the element for which you want to create a view.
- 3 Drag the icon onto the drawing page.

**NOTE** Any changes you make to the original element will also automatically appear in views of the element.

**To duplicate an existing element in a diagram:**

- 1 In the diagram, select the element you want to duplicate.
- 2 Press Ctrl+drag to create a copy of the element.

Two elements in the same package cannot have the same name, so when you duplicate an element, Visio Enterprise automatically gives the duplicate a new name.

**TIP** By default, pressing Ctrl+dragging an element creates a copy of the element. You can change the default behavior so that Ctrl+dragging creates a view of an element rather than a copy. To change the behavior, choose UML > Options, then, under Shape Ctrl-Drag Behavior on the UML Add-On tab, choose Copy Object View.

**To delete an element from a diagram:**

- 1 In the diagram, select the shape that represents the element you want to delete.
- 2 Choose Edit > Clear.
- 3 In the UML Shape Delete dialog box, choose the option you want, then click OK.

If you choose the first option, you delete not only the current view of the element but also the element itself and all of its other views. If you choose the second option, you delete only the current view of the element.

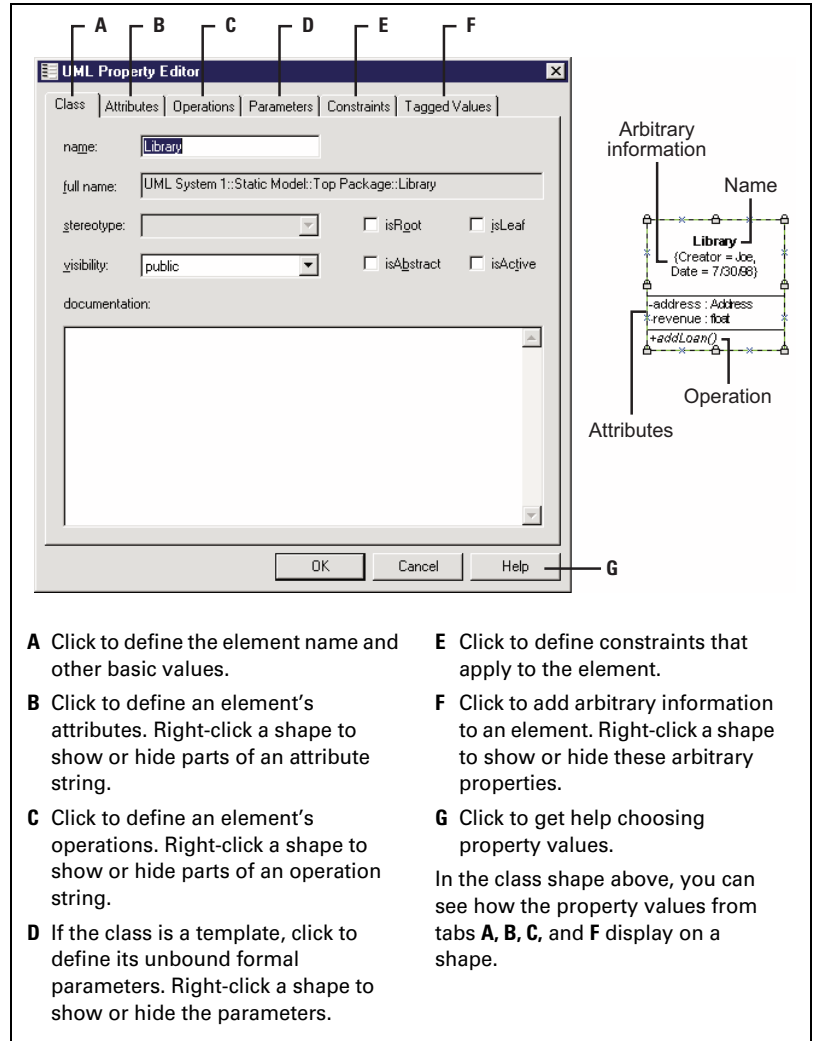
**TIP** You can change the default behavior for deleting an element from a diagram. To change the default, choose UML > Options. Under Shape Delete Options on the UML Add-On tab, choose the option you want, then click OK.

For more details about deleting UML elements, search online help.

## Associating property values with UML elements

A **property** is a value attached to a model element. For many elements, the UML specifies the values you can add. For example, classes can have attributes, operations, visibility, and more. You can also use a tagged value property to add arbitrary information to elements.

**Figure 5-2**  
UML Property Editor for a class



In Visio Enterprise, each element has an associated UML Property Editor, where you can quickly add values for both specified and arbitrary properties.

**To associate property values with an element:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, double-click the icon representing the element with which you want to associate properties. Or, on the drawing page, double-click the shape that represents the element.
- 3 In the UML Property Editor, add the property values you want on the first tab. Click additional tabs to add more values, then click OK.

**Diagnosing and fixing semantic errors**

As you create UML system models in Visio Enterprise, a semantic error checker runs in the background identifying and diagnosing errors, such as missing data and improper use of the notation. When the error checker identifies an error in an element, the shape representing the element in a diagram turns red. In addition, an error message appears in the Errors window of the UML Navigator.

For more details about semantic error checking, see [“Conforming to the UML syntax”](#) in [Chapter 4](#), [“Modeling software components.”](#)

**To diagnose and fix an error:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the UML Navigator, click the Errors tab to display the Errors window.  
Errors are listed according to error type. You can drag the border of the UML Navigator to make the Errors window wider, or use the scroll bar at the bottom of the Errors window to scroll to the right.
- 3 Read a description of the error to diagnose the problem.
- 4 Double-click an error to locate the icon or shape that represents the UML element you need to fix.

Visio Enterprise selects the icon representing the element in the tree view (the selected tree view icon turns gray). If the element with the error is represented by a shape in a UML diagram, Visio Enterprise displays the diagram and selects the shape (the shape has red error marking).

- 5 Using the information provided by the error description in the Errors window, fix the error.

# The UML diagrams

The UML notation supports an object-oriented approach to modeling systems. While it defines diagram types that are useful at various phases of the modeling process, the UML notation does not recommend when the diagrams should be used or what development process to follow.

As a UML notation tool, Visio Enterprise also supports the creation of models without recommending any particular development process. However, for purposes of showing the diagrams in the context of a meaningful system model, in this section the UML diagram types are organized into models that roughly correspond to phases a development process might follow:

- Phase 1: use case model
- Phase 2: domain model
- Phase 3: design model
- Phase 4: implementation model

While the phases are numbered consecutively, the most effective development processes are iterative, with each phase occurring a number of times.

**NOTE** The diagrams in the following sections, which illustrate a library information system, originally appeared in object-oriented analysis and design training materials written by Craig Larman for ObjectSpace ([www.objectspace.com](http://www.objectspace.com)). The activity, component, and deployment diagrams originally appeared in the *OMG Unified Modeling Language Specification*.

For details about how Visio Enterprise supports the creation of models and diagrams, see “[The Unified Modeling Language in Visio Enterprise](#)” in [Chapter 4](#), “[Modeling software components](#).”

## Phase 1: use case model

Use cases are narrative descriptions of processes that you create early in a development cycle. They describe the interactions between external actors and your system, and help you understand the system requirements and the terminology used in the domain area. During the use case phase, you create use case diagrams.

### *Use case diagrams*

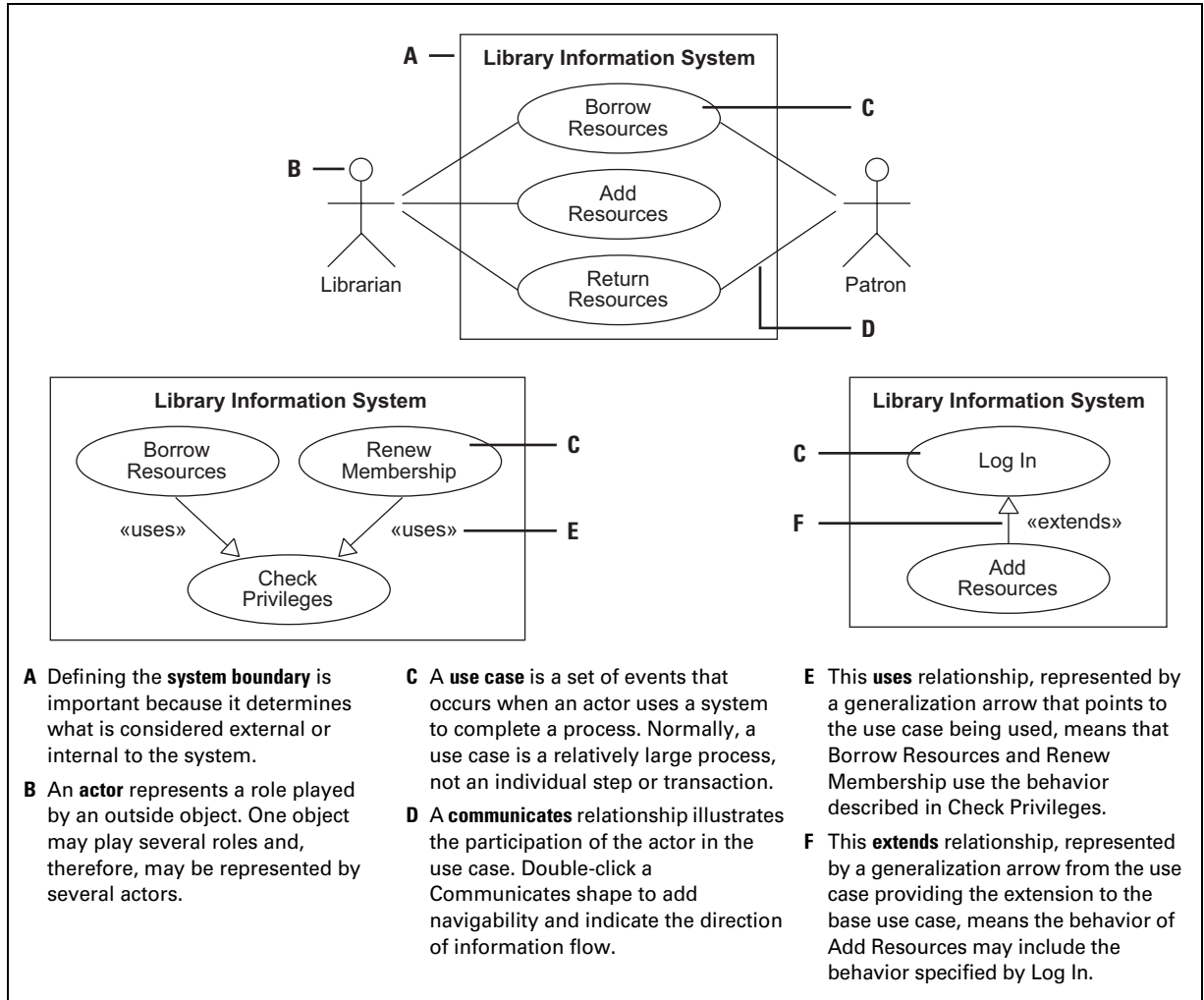
In the early stages of a development project, use case diagrams describe real-world activities and motivations. After you identify the use cases, you can create use case diagrams to put the use cases in context. Creating a use case diagram involves establishing a system boundary for a set of use cases and defining the lines of communication between a particular actor and a use case. You can refine the diagrams in later stages to reflect user interface and design details.

### To create a use case diagram:

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view or on the drawing page, right-click the package in which you want to include the diagram, then choose New > Use Case Diagram from the shortcut menu.

For more details about use case diagrams, right-click a use case shape, then choose Shape Help, or search online help.

**Figure 5-3** Use case diagram



## Phase 2: domain model

Like the use case phase, the domain model phase is focused on building an understanding of the domain for which you're developing a system. This is an analysis phase in which you're still thinking about objects and relationships in the real world rather than about programming concepts. During the domain model phase, you create conceptual static structure diagrams, package diagrams, and sequence diagrams.

### *Conceptual diagrams*

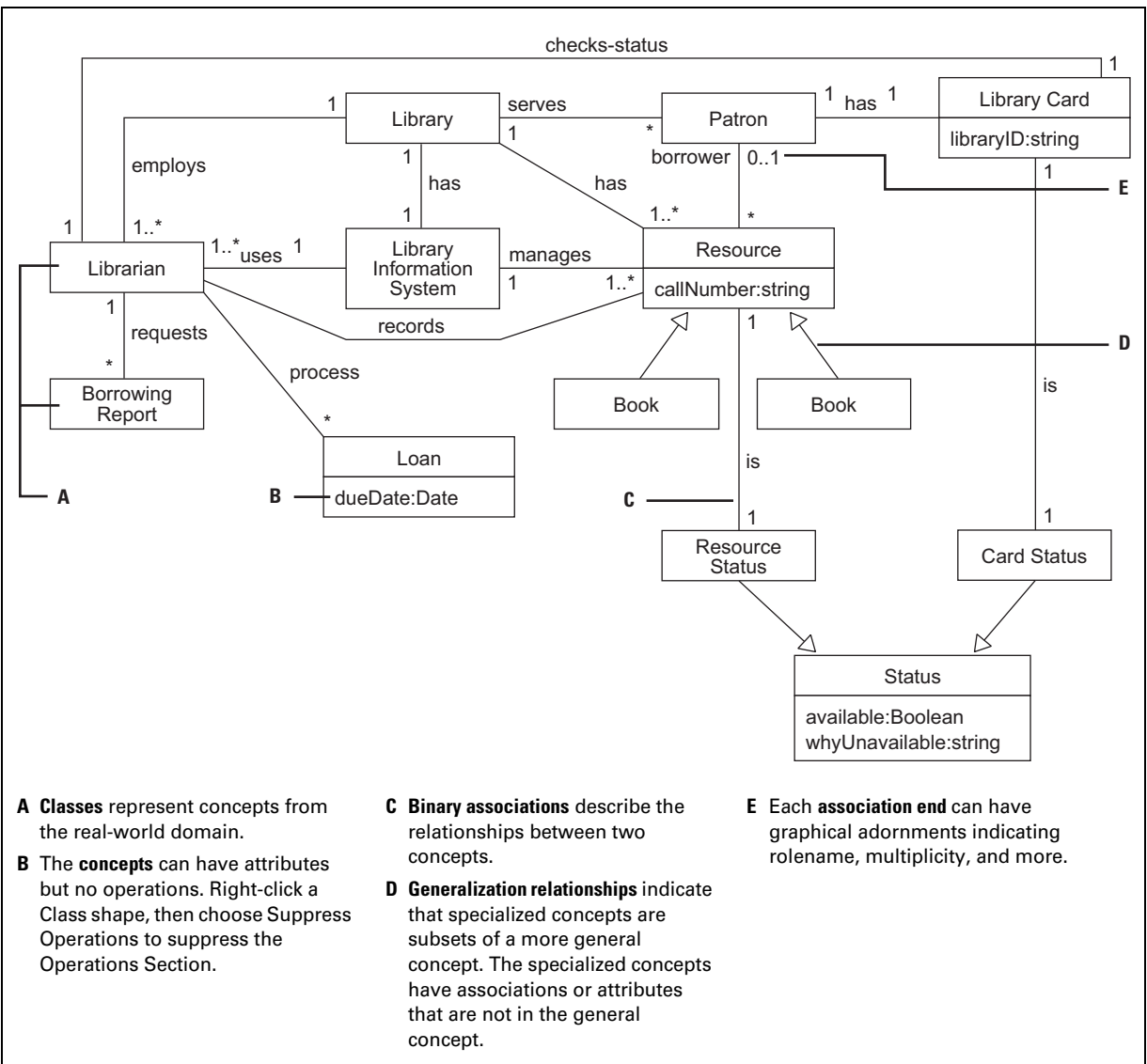
A conceptual diagram is a static structure diagram that represents concepts from the real world and the relationships between them. It is a model of meaningful ideas, things, and objects from the problem domain. The conceptual model focuses on relationships and attributes rather than methods, and helps you understand the terminology in the domain area for which you are developing a system.

#### **To create a conceptual diagram:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, right-click the icon representing the class or package in which you want to include the diagram, then choose **New > Static Structure Diagram** from the shortcut menu.

For details about static structure diagrams, right-click a static structure shape, then choose **Shape Help**, or search online help.

**Figure 5-4** Conceptual diagram



## *Package diagrams*

As your understanding of the problem domain increases, your conceptual model becomes more complicated and may reach an unmanageable size. You can use packages to divide your model into smaller, more manageable subsets. Or, you can use packages to define system architecture. A package is the basic organizing element of a UML system model. You can think of the entire system as a package that contains all the other packages, diagrams, and elements. One package can contain subordinate packages, diagrams, or single elements, and you can set the visibility of a package as well as the visibility of the elements it contains.

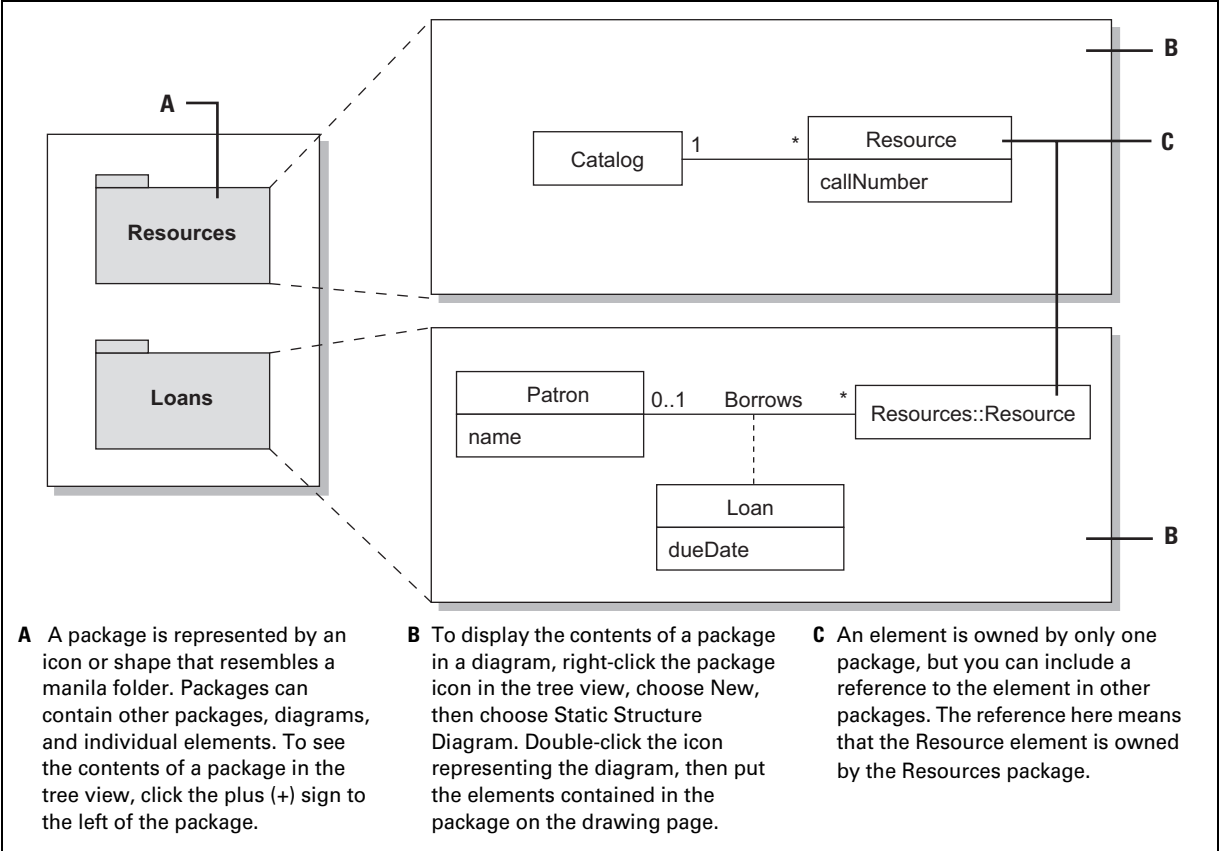
### **To create a package diagram:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, right-click the icon for the package you want to use as a container for other packages, then choose **New > Static Structure Diagram** from the shortcut menu.  
  
Visio Enterprise adds an icon representing the diagram to the tree view and automatically displays the UML Static Structure stencil and a blank drawing page.
- 3 Drop a Package shape onto the drawing page for each subpackage you want the top package to contain.  
  
Visio Enterprise automatically adds a package icon for each subpackage to the tree view.
- 4 In the tree view, right-click the icon for each of the subpackages contained within the top package, choose **New**, then choose the type of diagram you want each subpackage to contain.
- 5 Double-click the icon for each subpackage diagram, then drop icons from the tree view or shapes from stencils onto the drawing page to represent the elements you want the subpackage to contain.

For more details about packages and package diagrams, right-click a package shape, then choose **Shape Help**, or search online help.



**Figure 5-5** Package diagram



## *Sequence diagrams*

A type of interaction diagram, a sequence diagram shows the actors or objects participating in an interaction and the events they generate arranged in a time sequence. Often, a sequence diagram shows the events that result from a particular instance of a use case, but a sequence diagram can also exist in a more generic form.

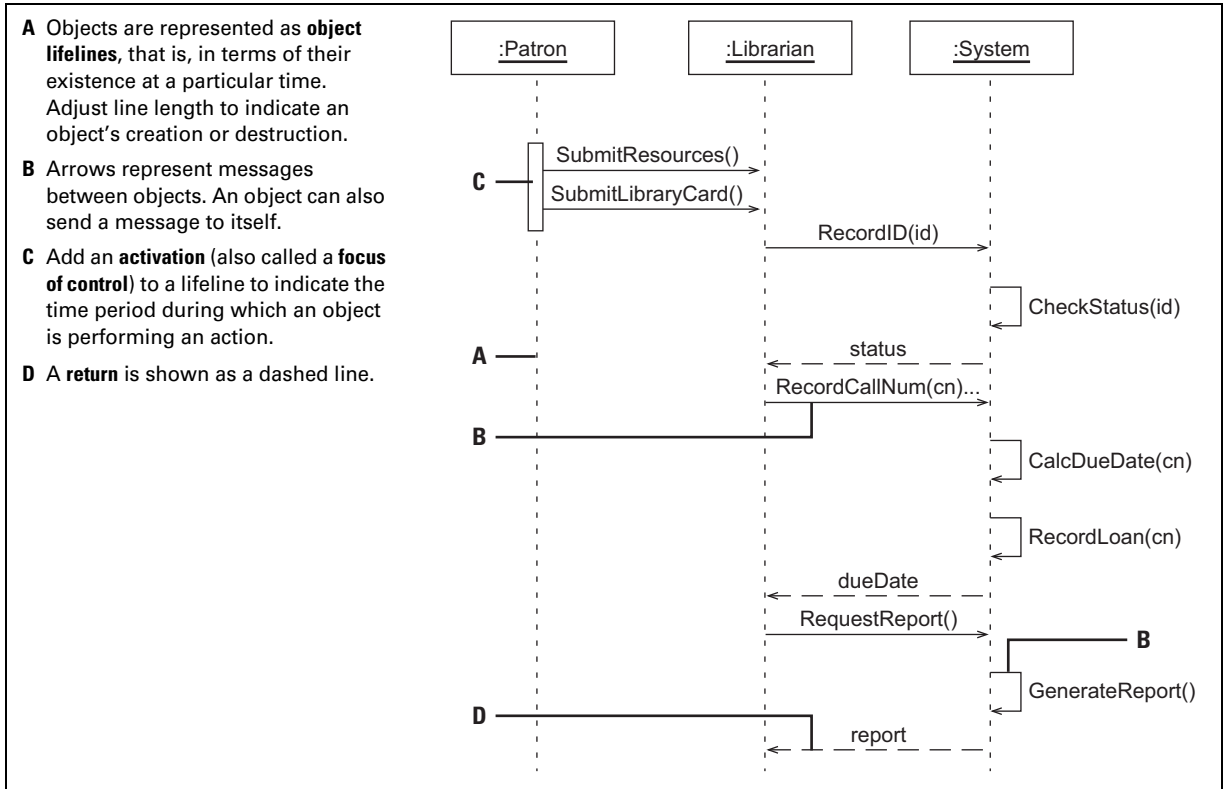
The vertical dimension in a sequence diagram represents time, with time proceeding down the page. The horizontal dimension represents different actors or objects.

### **To create a sequence diagram:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view or on the drawing page, right-click the package in which you want to include the diagram, then choose New > Sequence Diagram from the shortcut menu.

For details about sequence diagrams, right-click a sequence diagram shape, then choose Shape Help, or search online help.

**Figure 5-6** Sequence diagram



### Phase 3: design model

During the use case and domain model phases, you focus on understanding the requirements and concepts related to the system you are developing. In the design phase, you apply this understanding and come up with a programming solution. To develop this solution, you need to use collaboration diagrams to determine how objects will communicate and class diagrams to define the classes that you will implement in the software. To understand the life cycle of an object, you can also create a statechart or activity diagram in relation to a particular class, concept, or use case.

## Collaboration diagrams

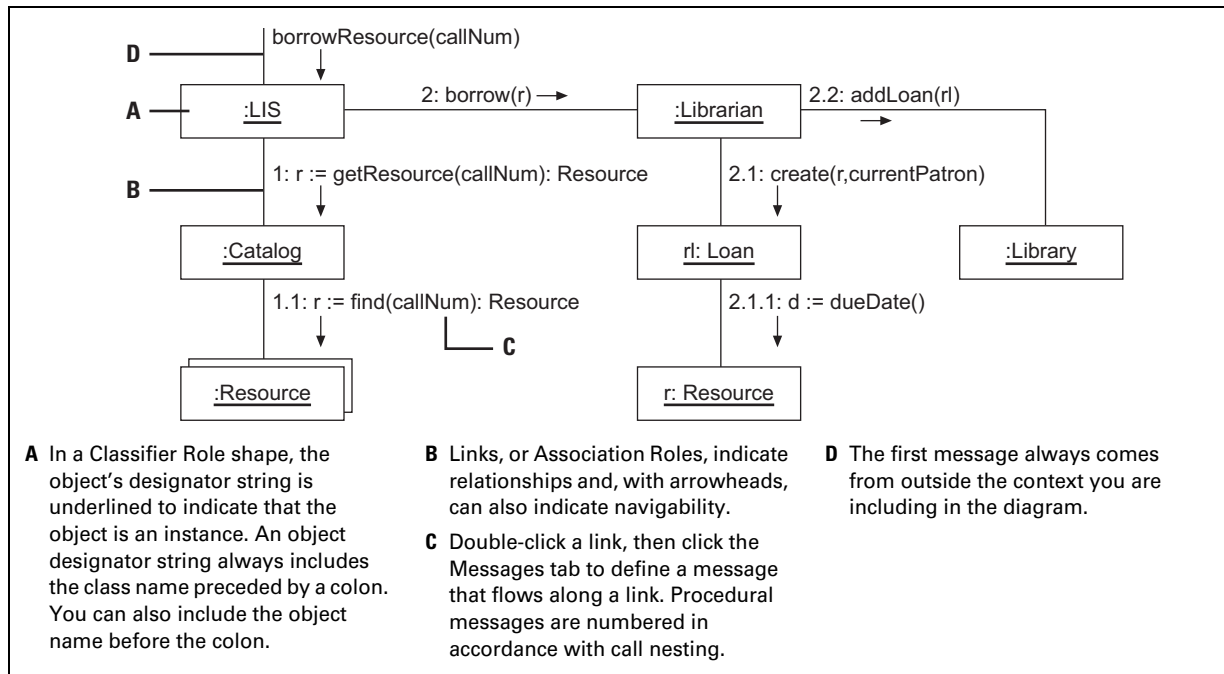
A collaboration diagram is an interaction diagram that shows, for one system event defined by one use case, how a group of objects collaborates with one another. Unlike a sequence diagram, a collaboration diagram shows relationships among object roles related in a particular context—a collaboration—and an interaction, which is the set of messages exchanged among the objects to achieve an operation or result. A collaboration diagram does not express time as a separate dimension. Therefore, the messages in a collaboration diagram are numbered to indicate their sequence.

### To create a collaboration diagram:

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view or on the drawing page, right-click the package in which you want to include the diagram, then choose **New > Collaboration Diagram** from the shortcut menu.

For details about collaboration diagrams, right-click a collaboration diagram shape, then choose **Shape Help**, or search online help.

**Figure 5-7** Collaboration diagram



## Class diagrams

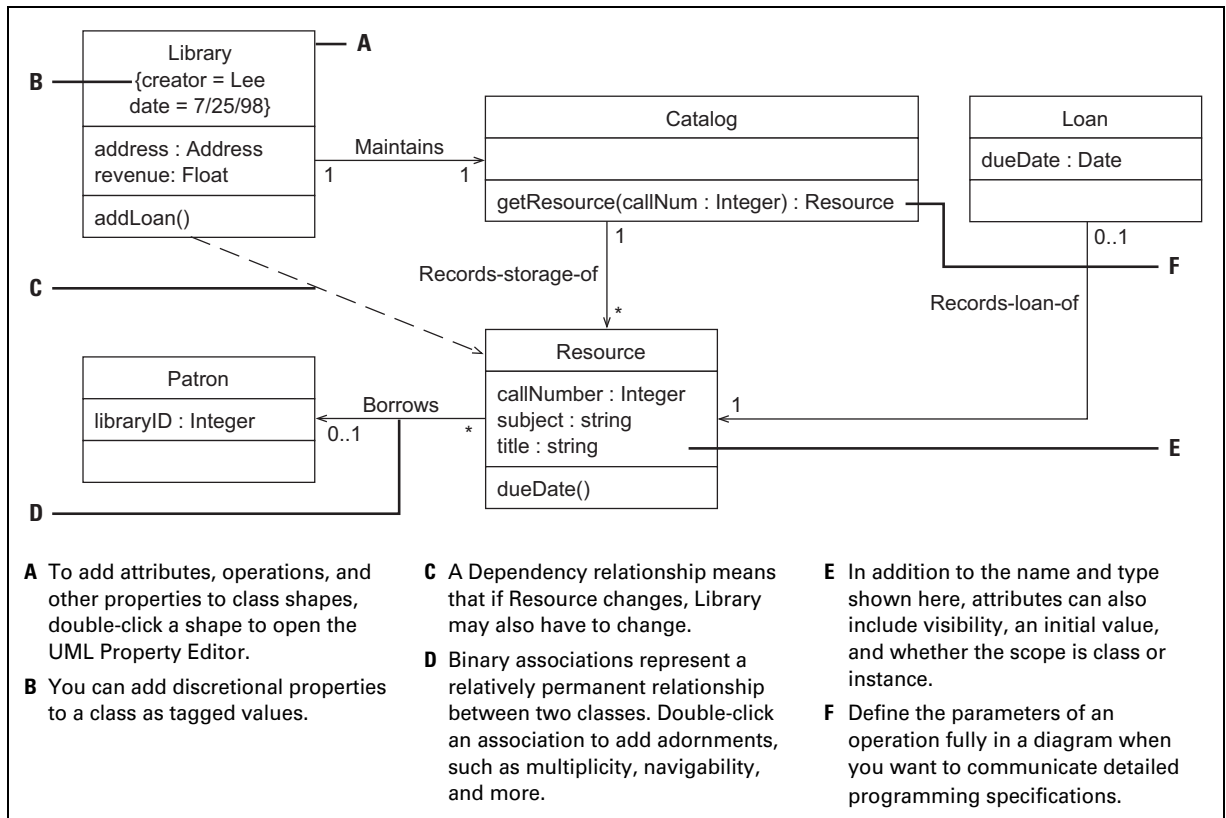
In the design phase, classes represent software entities, and class diagrams include class and interface specifications. Like conceptual diagrams, class diagrams are static structure diagrams that decompose a software system into its parts. In a class diagram, however, the parts are classes that represent fully defined software entities rather than objects that represent real-world concepts. In addition to attributes and associations, a class diagram also specifies operations, methods, interfaces, and dependencies.

### To create a class static structure diagram:

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view or on the drawing page, right-click the class or package in which you want to include the diagram, then choose New > Static Structure Diagram from the shortcut menu.

For details about static structure diagrams, right-click a static structure shape, then choose Shape Help, or search online help.

**Figure 5-8** Class diagram



## Statechart diagrams

A **state machine**, which is attached to a class or use case, is a graph of states and transitions that describes the response of an object to outside stimuli. A statechart diagram represents a state machine. By documenting events and transitions for a single class or use case, a statechart diagram shows the sequence of states a single object goes through during its lifetime.

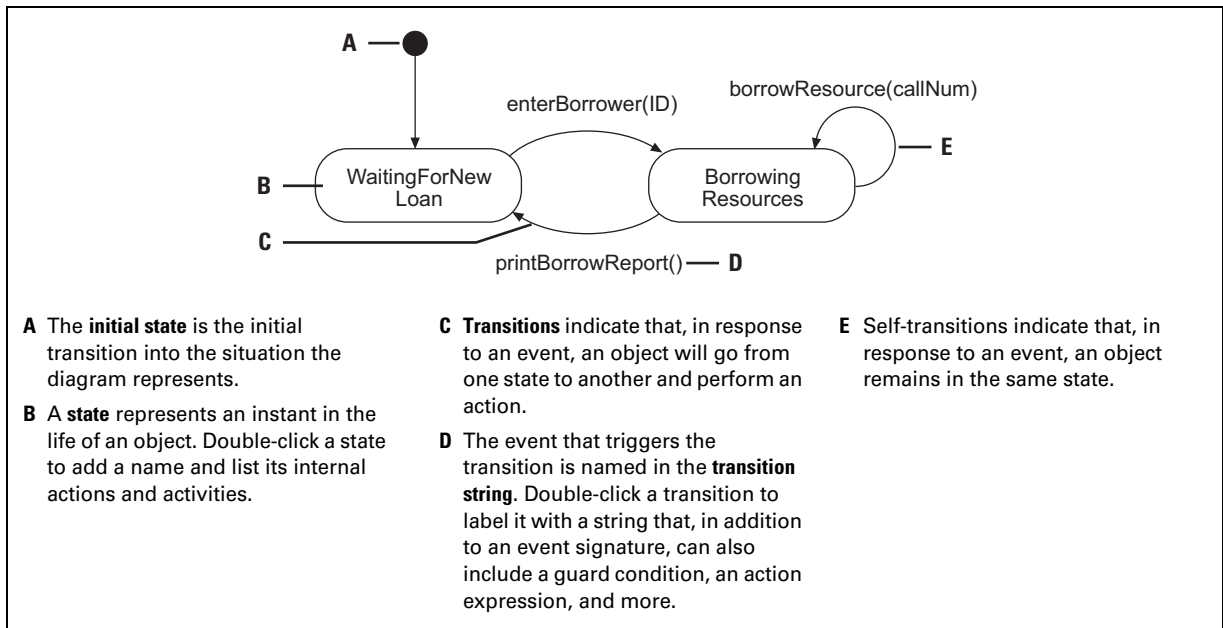
To represent a flow driven by internally generated actions rather than external events, use an activity diagram (for details, see the next section).

### To create a statechart diagram:

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, right-click the icon that represents the class or use case with states you want to document, then choose **New > Statechart Diagram** from the shortcut menu.

For details about statechart diagrams, right-click a statechart diagram shape, then choose **Shape Help**, or search online help.

**Figure 5-9** Statechart diagram



## *Activity diagrams*

An activity diagram is a special case of a statechart diagram in which all of the states are action states and the transitions are triggered by the completion of actions in the source state. Related to a specific class or use case, an activity diagram describes the internal behavior of a method. Activity diagrams encourage you to notice and document parallel and concurrent activities. This makes them excellent tools for modeling workflow, analyzing use cases, and dealing with multi-threaded applications.

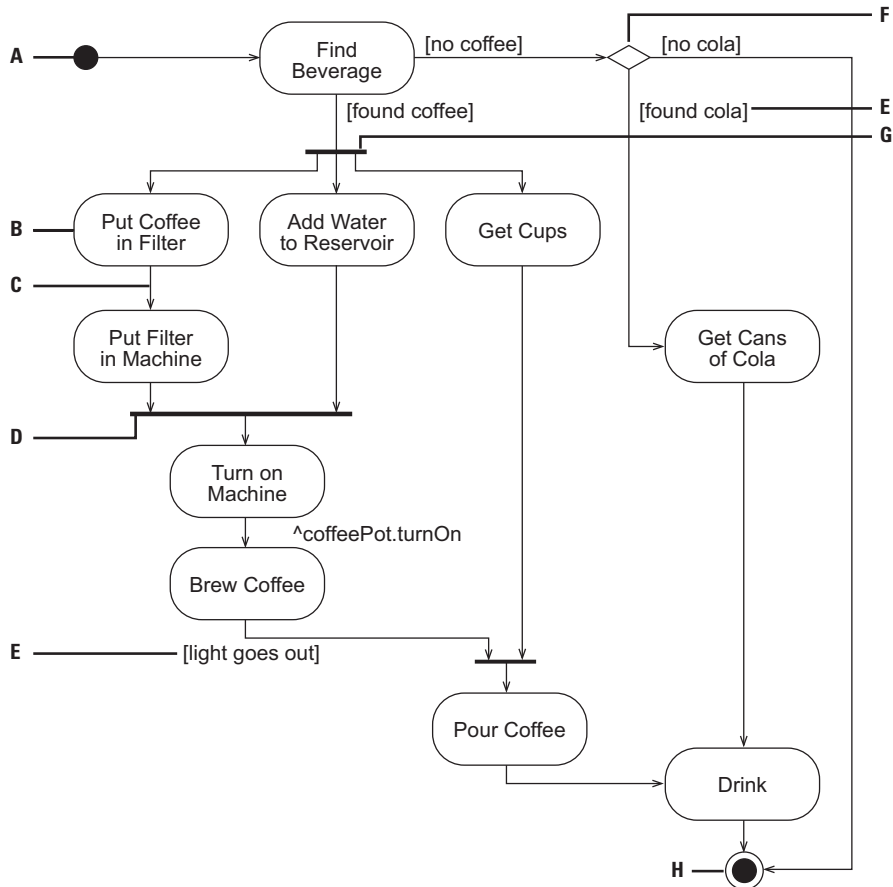
Use an activity diagram to represent a flow driven by internally generated actions. Use a statechart diagram to represent a flow in response to external events (for details, see the previous section).

### **To create an activity diagram:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view, right-click the class or use case icon that includes the state you want to represent, then choose New > Activity Diagram from the shortcut menu.

For details about activity diagrams, right-click an activity diagram shape, then choose Shape Help, or search online help.

**Figure 5-10** Activity diagram



**A** The **initial state** is the initial transition into the situation the diagram represents.

**B** An **action state** is a type of state that represents a completed activity.

**C** A **transition** from an action state occurs when the action state's internal action is complete.

**D** To indicate concurrent activities that must be completed before the next activity can occur, use a join transition.

**E** Double-click transitions from action states to label them with guard conditions and action expressions.

**F** When one decision leads to another, use a diamond shape to indicate the second, nested decision.

**G** To indicate activities that can occur in parallel, use a fork transition.

**H** The **final state** represents the completion of activity in the situation the diagram represents.



## Phase 4: implementation model

The implementation phase focuses on the physical and component structures of the development environment. During the implementation phase, you create component diagrams and deployment diagrams. A component diagram shows the dependencies among software components. A deployment diagram shows the configuration of the run-time processing elements and the software components, processes, and objects that live on them.

### *Component diagrams*

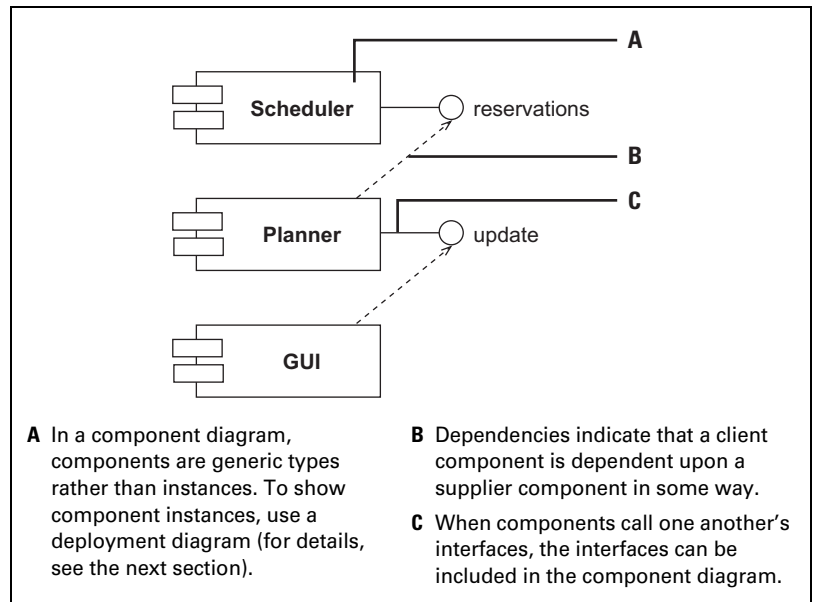
Component diagrams are implementation-level diagrams that show the structure of the code itself. A component diagram consists of components, such as source code files, binary code files, executable files, or dynamic-link libraries (DLLs), connected by dependencies. Use a component diagram to partition a system into cohesive components. Typically, each component in a component diagram is documented in more detail in a use case or class diagram.

#### **To create a component diagram:**

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view or on the drawing page, right-click the package in which you want to include the diagram, then choose **New > Component Diagram** from the shortcut menu.

For details about component diagrams, right-click a component diagram shape, then choose **Shape Help**, or search online help.

**Figure 5-11**  
Component diagram



## Deployment diagrams

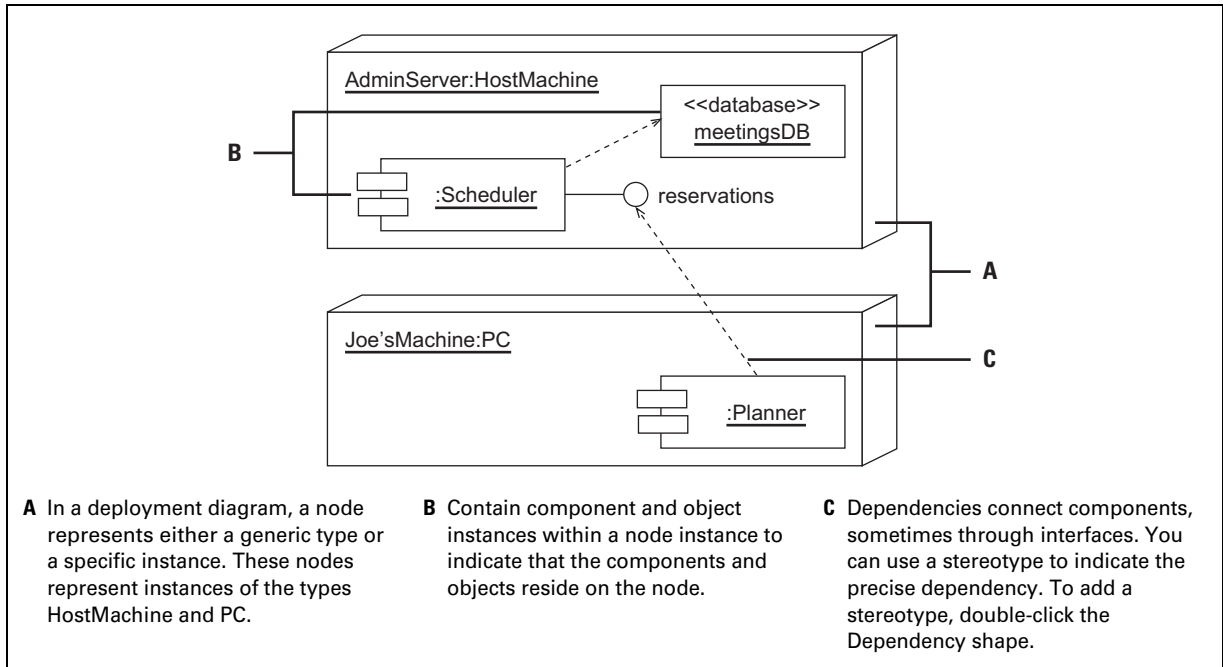
Deployment diagrams are implementation-level diagrams that show the structure of the run-time system. From a deployment diagram, you can understand how the hardware and software elements that make up an application will be configured and deployed. Deployment diagrams consist of nodes, components, and the relationships between them.

### To create a deployment diagram:

- 1 Open the UML Model Diagram file that represents your system.
- 2 In the tree view or on the drawing page, right-click the package in which you want to include the diagram, then choose New > Deployment Diagram from the shortcut menu.

For details about deployment diagrams, right-click a deployment diagram shape, then choose Shape Help, or search online help.

**Figure 5-12** Deployment diagram



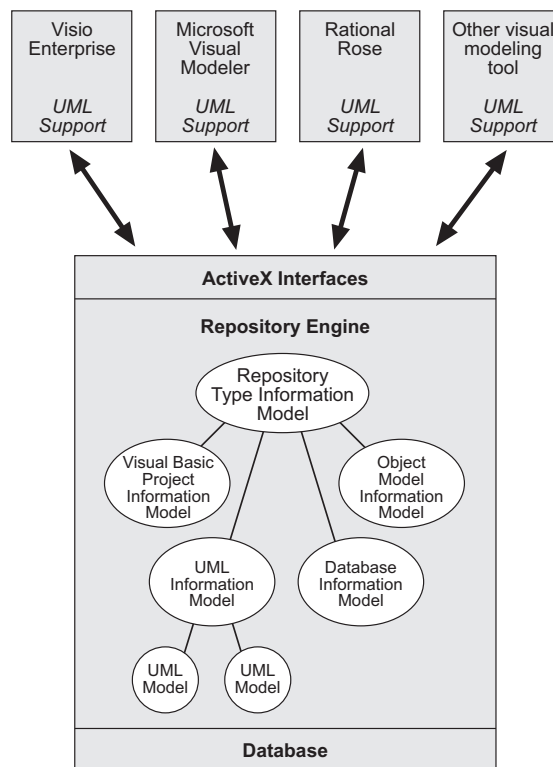
# Storing and retrieving models: working with Microsoft Repository

Through its UML solution, Visio Enterprise supports the sharing and exchange of software models. When you install Visio Enterprise, you also install and register all the components you need to store models in and retrieve them from Microsoft Repository 2.0. Microsoft Repository is a storage mechanism that consists of ActiveX interfaces developers can use to define information models, a repository engine that is the underlying storage mechanism for the models, and a database. Using the Repository Type Information Model stored in the repository engine, you can create specific tool information models, which define the objects a model uses. When software components are stored as information models, a developer can access and use the components regardless of which tool the components were created in or which platform they run on.

The following illustration shows the relationships among Visio Enterprise, Microsoft Repository, and other modeling solutions.

**Figure 5-13**

The first time you export UML static structure data, Visio Enterprise creates the UML information model in Microsoft Repository that defines the objects UML models can use. Through Microsoft Repository, Visio Enterprise can exchange UML models with any other visual modeling program that also supports the UML information model.



## Importing data from Microsoft Repository

You can import a UML static structure model from Microsoft Repository regardless of which tool originally created the model. When you import a model, it appears in the form of icons in the UML Navigator tree view. You can drag the icons onto the drawing page to create static structure diagrams and you can access the properties of a UML element by double-clicking an icon in the tree view or a shape on the drawing page.

### To import a model from Microsoft Repository:

- 1 Choose File > New > UML > UML Model Diagram to open the UML Model Diagram template.
- 2 Choose UML > Import.
- 3 In the UML Import dialog box, type the name and location of, or click Browse to locate, the Microsoft Repository database that includes the model you want to import. If necessary, type your user name and password.
- 4 Click List Models, then select the name of the model you want to import.  
Visio Enterprise imports the model and adds icons to the existing system in the tree view.

**TIP** If you import a model into a new drawing file, you can delete the default static model after the new data is imported. To delete the static model, right-click the model's icon, then choose Delete.

## Exporting data to Microsoft Repository

Currently, the Microsoft Repository Type Information Model supports the creation of UML information models only for static structure data, such as the data found in conceptual, object, or class diagrams. Therefore, only the static structure components of your Visio Enterprise UML model can be exported to Microsoft Repository. When you export, property data you have added to the UML elements is exported along with the elements themselves.

For details about adding properties to UML elements, see [“Associating property values with UML elements”](#) earlier in this chapter.

When you choose the Export command, the UML solution’s semantic error checking automatically checks for errors that will cause problems in the static structure data you export. Errors are listed in the Errors window in the UML Navigator.

For details about semantic error checking, see [“Diagnosing and fixing semantic errors”](#) in this chapter and [“Conforming to the UML syntax”](#) in [Chapter 4](#), [“Modeling software components.”](#)

### To export data to Microsoft Repository:

- 1 Open the UML Model Diagram file that represents the system that includes the static structure data you want to export, then choose UML > Export.
- 2 In the Export dialog box, type or browse for the name of an existing database or type an ODBC data source name or connection string.
- 3 If necessary, type your user name and password.
- 4 Click List Models to list the existing models in Microsoft Repository.
- 5 Select the name of the model you want to replace or type a new model name, then click OK.

**TIP** You can also export static structure data by right-clicking the UML System icon in the tree view, then choosing Export from the shortcut menu.

# Reverse engineering source code

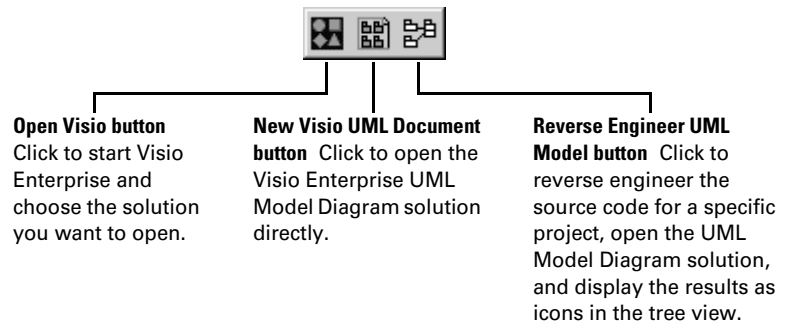
If you create Microsoft Visual C++ or Microsoft Visual Basic code, you can reverse engineer your projects' class definitions to generate UML static structure models in Visio Enterprise. From the tree view model generated in the Visio Enterprise UML Navigator, you can drag icons onto a drawing page to create static structure diagrams, and you can access the properties of UML elements by double-clicking an icon in the tree view or a shape on the drawing page.

For details about UML models and diagrams, see “[The Unified Modeling Language in Visio Enterprise](#)” in [Chapter 4](#), “[Modeling software components](#).”

When you install Visio Enterprise, you also install DLL files, which you can use to customize Microsoft Visual C++ 5.0 and 6.0 and Microsoft Visual Basic 5.0 and 6.0.

**Figure 5-14**

When you customize Microsoft Visual C++ and Microsoft Visual Basic using the Visio Enterprise add-ins, Visio Enterprise creates a new three-button toolbar.



## Microsoft Visual C++

The Microsoft Visual C++ Visio UML Add-in uses Browse Information files to create UML models. Therefore, you must generate a Browse Information file when you compile a build for the project with source code you want to reverse engineer.

### To customize Microsoft Visual C++ with the UML add-in:

- 1 Copy the BSKKIT50.exe or BSKKIT60.exe file into your \Visio\Solutions\UML folder.

To locate BSKKIT50.exe, connect to <http://support.microsoft.com/support/kb/articles/Q153/3/93.asp?PR=CHS&T1=7d&FR=0&A=T&T=B&S=F&>. You will be prompted to supply information as a new user or your e-mail address and password if you've already registered. For additional information, contact Microsoft Technical Support.

To locate BSCKIT60.exe, you must have access to a Microsoft Developer Network (MSDN) Professional or Universal subscription. The file is on Development Platform (US) Disk 2 in the UNSUP-ED folder.

- 2 In Microsoft Visual C++, choose Tools > Customize.
- 3 In the Customize dialog box, click the Add-ins And Macros tab, then click Browse.
- 4 In the Browse dialog box, for Files Of Type, choose Add-ins (.dll).
- 5 Locate the appropriate UML add-in, called Umlvc60.dll or Umlvc50.dll, which is located in the \Visio\Solutions\UML folder, then click Open.  
Visio UML Add-in appears in the list of add-ins on the Add-ins And Macros tab with a check mark beside it.
- 6 Click Close.  
The floating, three-button Visio Enterprise toolbar appears.
- 7 Dock the toolbar by dragging it into the toolbar area if you want the toolbar to be available each time you open Microsoft Visual Studio.

**TIP** If you want to create keyboard shortcuts for each of the Visio Enterprise toolbar buttons, choose Tools > Customize, click the Keyboard tab, then choose the Add-in category.

#### **To generate a Browse Information file:**

- 1 In Microsoft Visual C++, choose Project > Set Active Project to set the project with source code you want to reverse engineer as the active project.
- 2 Choose Project > Settings.
- 3 In the Project Settings dialog box, choose the type of build configuration you want, click the C/C++ tab, then check Generate Browse Info.
- 4 Click the Browse Info tab to specify the name and location of the Browse Information file, then click OK.

Keep the default file name (which is the project name with a .bsc extension). If you place the Browse Information file in the root directory for your project, the Visio UML Add-in automatically finds the file. If you place the file in a subfolder within the directory, you will have to browse to locate the file at the time you reverse engineer the source code.

### To reverse engineer source code in Microsoft Visual C++:

- 1 Customize Microsoft Visual C++ according to the procedure [“To customize Microsoft Visual C++ with the UML add-in:”](#) on page 102.
- 2 Click the Reverse Engineer UML Model button on the Visio Enterprise toolbar.
- 3 In the Select Project dialog box, check the projects you want to reverse engineer, then click Create Model.

It may take several seconds to extract the class information from the Browse Information file. When the extraction is complete, the Visio Enterprise UML Model Diagram solution opens with a blank static structure diagram drawing page, and a tree view in the UML Navigator populated with icons that reflect the class definitions in the source code.

- 4 In Visio Enterprise, drag icons from the tree view onto the drawing page to create a static structure diagram that represents a view of the model.

For details about creating UML diagrams, see [“How to model a system using the UML solution”](#) earlier in this chapter.

## Microsoft Visual Basic

The Visio UML Add-in creates a toolbar in Microsoft Visual Basic that you can use to reverse engineer source code to create a UML static structure model in Visio Enterprise. Open the Visual Basic project that contains the code you want to reverse engineer before you click the toolbar button.

### To customize Microsoft Visual Basic with the UML add-in:

- 1 In Microsoft Visual Basic, choose Add-ins > Add-in Manager.
- 2 Depending on the version of Visual Basic you are using, do the following:
  - In Visual Basic 5.0, in the Add-in Manager dialog box, check Visio UML Add-in, then click OK. The three-button Visio Enterprise toolbar appears.
  - In Visual Basic 6.0, in the Add-in Manager dialog box, select Visio UML Add-in. For Load Behavior, check Loaded/Unloaded and Load On Startup, then click OK. The three-button Visio Enterprise toolbar appears.

**TIP** When the Visio UML Add-in creates the toolbar, it also adds commands to the Tools menu. You can locate the commands by choosing Tools > Visio Solution.



**To reverse engineer source code in Microsoft Visual Basic:**

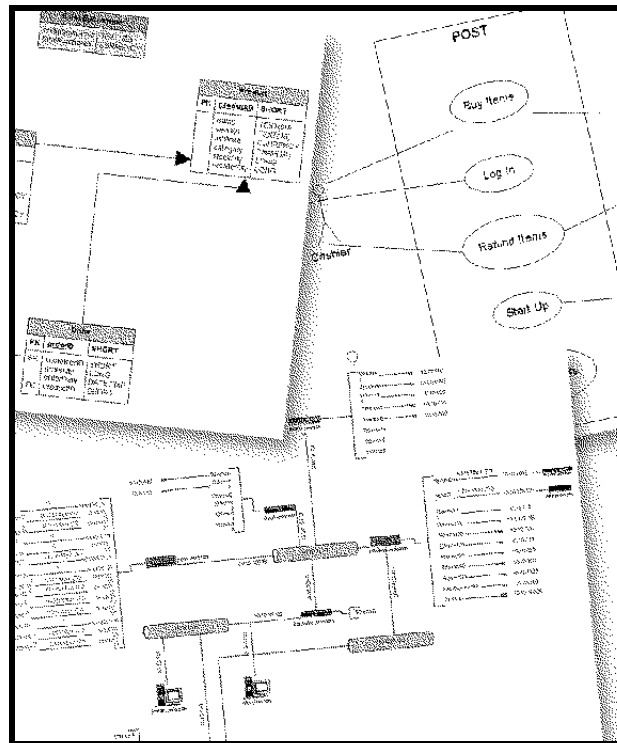
- 1** Customize Microsoft Visual Basic according to the previous procedure, and open the project that contains the source code you want to reverse engineer.
- 2** Click the Reverse Engineer UML Model button on the Visio Enterprise toolbar.

The Visio Enterprise UML Model Diagram solution opens with a blank static structure diagram drawing page, and a tree view in the UML Navigator populated with icons that reflect the class definitions in the source code.

- 3** In Visio Enterprise, drag icons from the tree view onto the drawing page to create a static structure diagram that represents a view of the model.

For details about creating UML diagrams, see [“How to model a system using the UML solution”](#) earlier in this chapter.





# Modeling a Database with Visio Enterprise



# Understanding the database modeling process

Visio® Enterprise provides powerful database modeling tools you can use to create logical models for relational and object-relational databases. A well-designed logical database model can help you ensure that any information system you design is accurate, complete, and efficient by showing in a visual format the relationships between the tables of information stored in the database. With Visio Enterprise, you can reverse engineer a model from an existing database, update a model based on changes in the database, generate a new database from the model, and update or forward engineer an existing database based on changes in the model.

This chapter introduces the Visio Enterprise database modeling tools and provides an overview of the database modeling process. For details about the topics introduced in this chapter, see “[Next steps](#)” at the end of this chapter.

## Topics in this chapter

- About the database modeling process in Visio Enterprise ..... 110
- Creating a database model ..... 116
- Refining a database model ..... 120
- Connecting to a database ..... 124
- Generating a database ..... 124
- Updating a model or a database ..... 125
- Creating a report ..... 127
- Next steps ..... 128

# About the database modeling process in Visio Enterprise

As an experienced database designer, you may work with other Visio products. If so, many of the tools in Visio Enterprise will be familiar. Other tools, such as the Types window, Tables & Views window, Code window, and Output window, may be new to you. This section introduces the basic process for modeling databases with Visio Enterprise and the user interface elements you'll use.

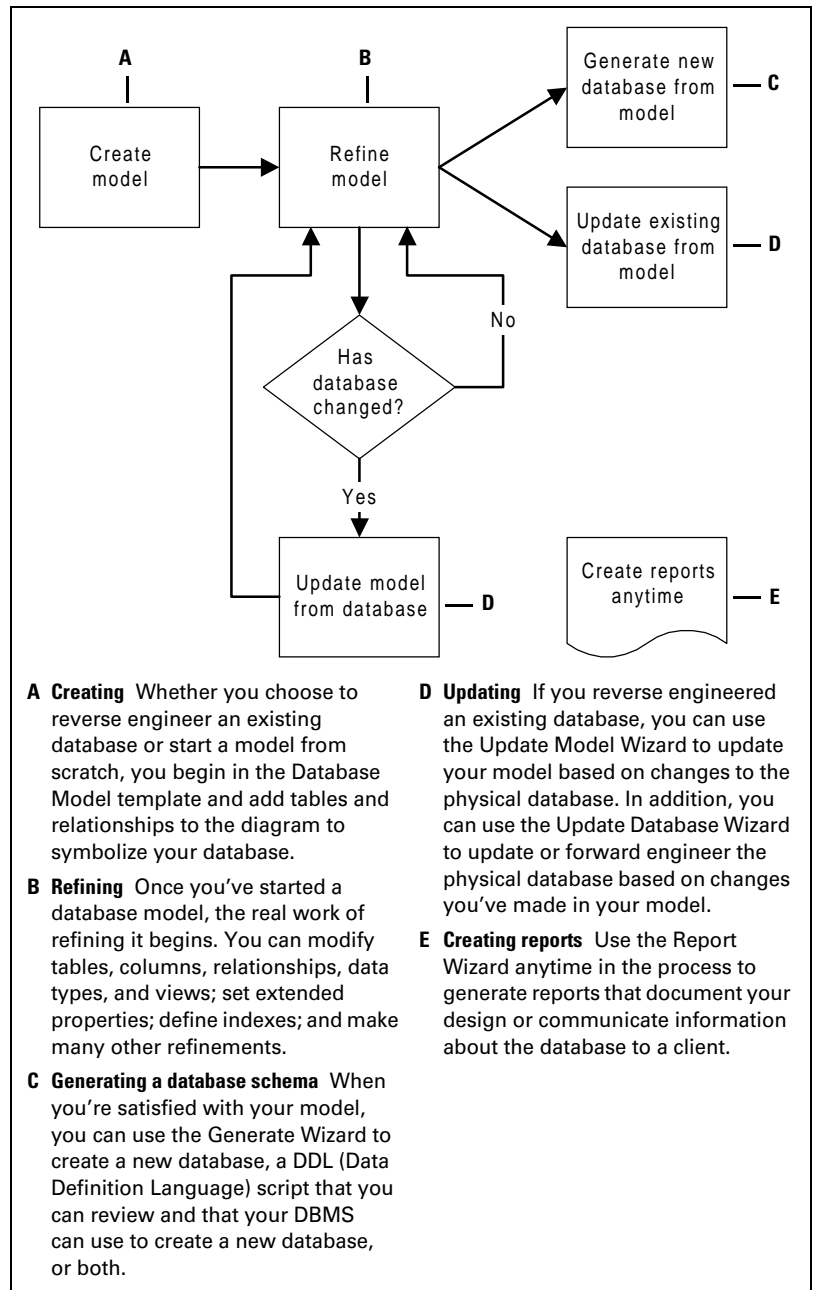
In addition to providing tools for creating database model diagrams, Visio Enterprise delivers functionality for connecting to a wide range of database management systems (DBMSs). As part of this functionality, Visio Enterprise offers workgroup-related support using Microsoft Repository.

Depending on the options you selected, the Visio Enterprise Setup program installs three templates in the \Visio\Solutions\Database folder: Bachman, Object Role Modeling, and Database Model. This chapter and subsequent chapters in this manual focus on the Database Model template. For details about the Bachman and Object Role Modeling templates, see online help.

When you model a database in Visio Enterprise using the Database Model template, your process may include some or all of the steps illustrated in the following figure.

**Figure 6-1**

Sample process for creating a database model



The Visio Enterprise drawing page contains the essential visual expression of a database model. The shapes in the diagram show the elements you want the database to store information about and make the relationships among them visible. Other information about your database, such as code for stored procedures, can also be included in the model, though it may not be visible in the diagram.

**Figure 6-2** The Visio Enterprise user interface

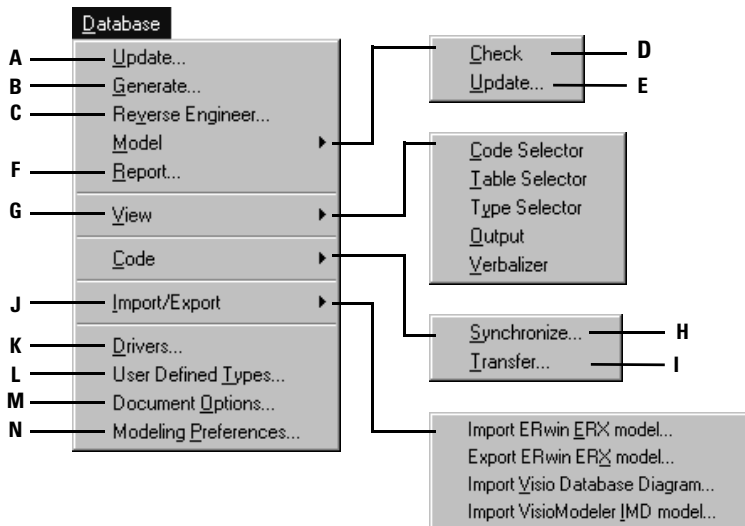




## Using the Database menu

When you start Visio Enterprise and open the Database Model template, the Database menu appears with commands that are unique to working in the database modeling solution. You also can right-click any model component to access component-specific commands.

**Figure 6-3** The Visio Enterprise Database menu and commands



**A Update** Opens the Update Database Wizard, so you can update an existing database to reflect your model.

**B Generate** Opens the Generate Wizard, so you can generate a new database directly or with a DDL script containing DBMS-specific commands.

**C Reverse Engineer** Opens the Reverse Engineer Wizard, so you can extract a model from an existing database.

**D Model > Check** Validates your model and displays any errors and warnings in the Output window.

**E Model > Update** Opens the Update Model Wizard, so you can update your model from a database.

**F Report** Opens the Report Wizard, so you can choose one of three report types to communicate information about your model.

**G View** Provides commands to open the Code window, Tables & Views window, Types window, Output window, and Verbalizer window.

**H Code > Synchronize** Keeps code that is saved externally in a mirror file in sync with the code in the model.

**I Code > Transfer** Transfers code for triggers, procedures, or check clauses into a DBMS-specific format.

**J Import/Export** Provides options to import existing database models from PLATINUM ERwin, Visio® Professional, and VisioModeler™ and to export models to ERwin and VisioModeler.

**K Drivers** Sets default drivers and mapping preferences, and defines options for generating DDL scripts.

**L User Defined Types** Opens the User Defined Types dialog box, in which you can create user-defined data types.

**M Document Options** Opens the Database Document Options dialog box, in which you can choose the default notation and display options for the current database model.

**N Modeling Preferences** Opens the Database Modeling Preferences dialog box, in which you can set default database modeling behavior that persists between Visio Enterprise sessions.

## Working with Microsoft Repository

Through its Database Model solution, Visio Enterprise supports sharing and exchange of database models. When you install Visio Enterprise, you also install and register all the components you need to store models in and retrieve them from Microsoft Repository.

Microsoft Repository is a storage mechanism that consists of ActiveX interfaces that developers can use to define information models, a Repository Engine that is the underlying storage mechanism for the models, and either a SQL Server or Microsoft Jet database in which the schema is stored. Microsoft Repository uses the Database Information Model (DBM) as a format for storing a database's data requirements independent of the physical implementation of the database. This common format enables you to easily share and update components in a team setting. When a database schema is stored as an information model, developers can access and use the components regardless of which tool the components were created in or on which platform they run.

To store your Visio Enterprise database model in Microsoft Repository, you establish a connection with the DBM Repository driver Visio Enterprise provides. The DBM Repository driver extracts the information from the model and converts it to a format that Microsoft Repository can read. The DBM Repository driver communicates with the Repository Engine, which manages the model data in Microsoft Repository. Your database model is then stored in a Repository database as a set of objects and the relationships between them.

**NOTE** Although the Repository database is stored in either a SQL Server or Microsoft Jet database, you can extract your database model from Microsoft Repository and specify any Visio Enterprise–supported database platform for implementation. The storage mechanism for Microsoft Repository is independent of the implementation of your database model.

The following figure illustrates the Repository architecture. Visio Enterprise supports Microsoft Repository version 2.0.

**Figure 6-4**

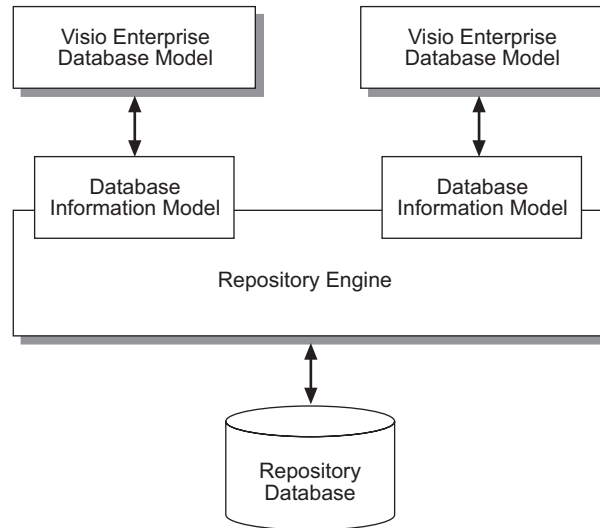
The Repository architecture

**Visio Enterprise Database Model**  
Model your database schema in Visio Enterprise.

**Database Information Model** Create a connection with the Microsoft Repository Database Information Model to extract the objects and relationships in a format Microsoft Repository can read.

**Repository Engine** Communicates with the DBM and manages the data in Microsoft Repository. It also provides the functions for storing the database model in the Repository database.

**Repository Database** A SQL Server or Microsoft Jet database in which the database model structure is stored.



For details about connecting to Microsoft Repository, see [Chapter 10, “Connecting to a DBMS.”](#) For details about working with Microsoft Repository in Visio Enterprise, see [Chapter 7, “Creating database models,”](#) [Chapter 11, “Generating a database schema,”](#) and [Chapter 12, “Updating models and databases.”](#)

# Creating a database model

When you want to create a database model in Visio Enterprise, you begin by opening the Database Model template. The Database Model template contains tools and commands to create new database models, and to connect to a DBMS to generate new or update existing databases.

The Database Model template opens the Entity Relationship stencil, the Object Relational stencil, and a blank drawing page. Once you've opened a template, you can take one of three approaches to start a database model in Visio Enterprise.

- **Create a new model from scratch.** Visio Enterprise provides the tools and controls you need to plan and design complex databases.
- **Use the Reverse Engineer Wizard to extract a model from an existing physical database.** Reverse engineering is the most widely used method for creating a database model, as it allows you to improve, enhance, and extend an organization's existing databases. You can choose to extract code stored as part of the database when you reverse engineer a database, and you can review and edit the code as you refine your model.
- **Import an existing model from PLATINUM ERwin, Visio Professional, or VisioModeler.** Importing an existing database model also saves time by building on existing information.

The following figure illustrates the three methods you can use to start a database model in Visio Enterprise.

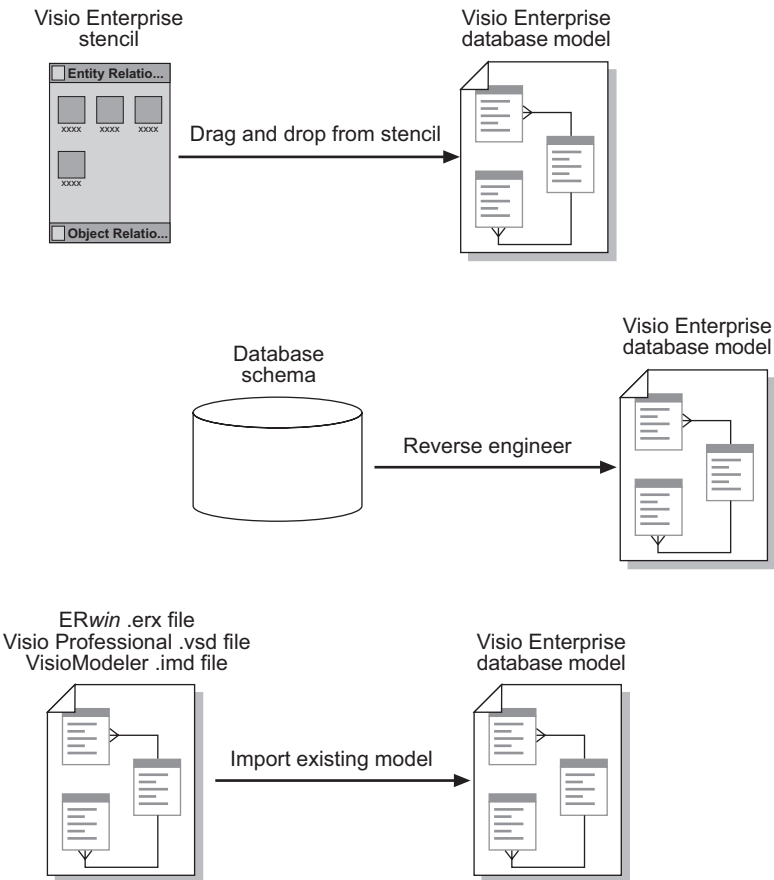
**Figure 6-5**

To create a database model, you can:

**Drag and drop shapes from the stencils** to create a new model from scratch.

**Use a reverse engineered model as a starting point** for either a new database or to make changes to the existing database.

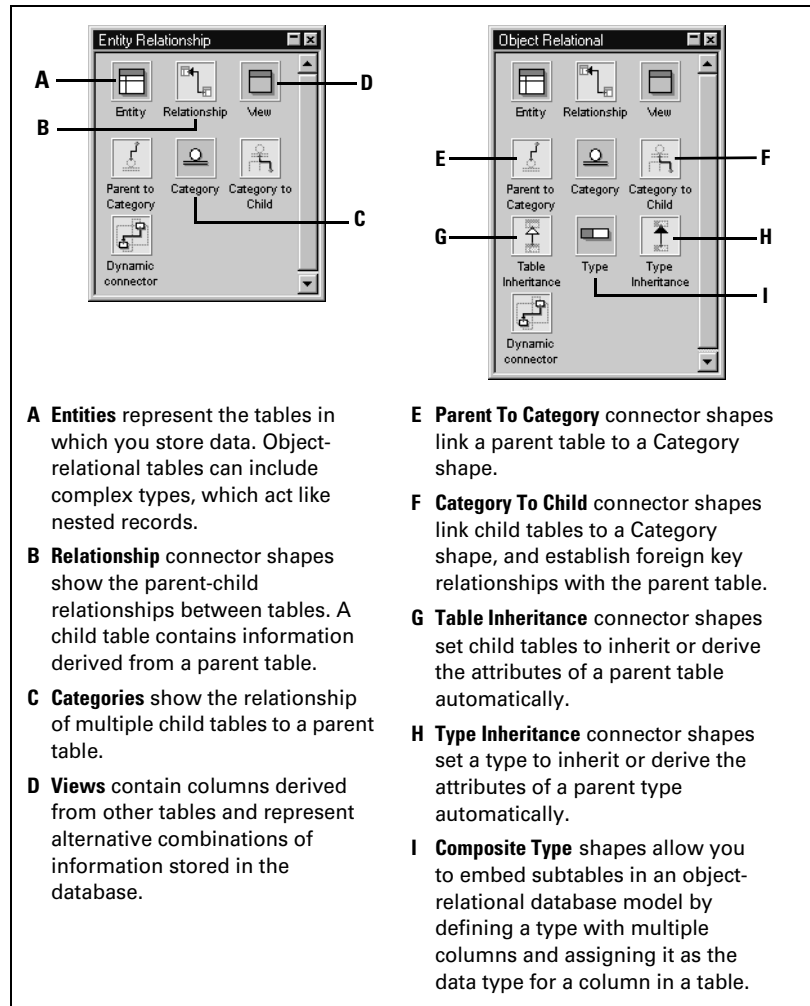
**Import an existing model** from PLATINUM ERwin, Visio Professional, or VisioModeler and refine it in Visio Enterprise.



The stencils in the Visio Enterprise Database Model template include SmartShapes® symbols, which are programmed to behave in very specific ways in accordance with the chosen modeling method. Shapes in the Entity Relationship stencil can be used to model relational databases. Shapes in the Object Relational stencil can be used to model either relational or object-relational databases.

**Figure 6-6**

Choose to use shapes from the Entity Relationship or Object Relational stencil based on the type of DBMS you will use to create the physical database.



**To reverse engineer a database model:**

- 1 In Visio Enterprise, choose File > New > Database > Database Model.  
Visio Enterprise creates a new, empty drawing page and opens the Entity Relationship and Object Relational stencils.
- 2 Choose Database > Reverse Engineer to start the Reverse Engineer Wizard.
- 3 Follow the instructions in the wizard.  
For details about reverse engineering and using the Reverse Engineer Wizard, see [“Reverse engineering an existing database”](#) in Chapter 7, “Creating database models.”

**To import a database model:**

- 1 In Visio Enterprise, choose File > New > Database > Database Model.  
Visio Enterprise creates a new, empty drawing page and opens the Entity Relationship and Object Relational stencils.
- 2 Choose Database > Import/Export, then choose the command you want.
- 3 In the Import dialog box, select a file to import, then click OK.  
For details about importing a database model from another application, see [“Importing a data model”](#) in Chapter 7, “Creating database models.”

**To create a new database model:**

- 1 In Visio Enterprise, choose File > New > Database > Database Model.  
Visio Enterprise creates a new, empty drawing page and opens the Entity Relationship and Object Relational stencils.
- 2 From the Entity Relationship or Object Relational stencil, drag table, relationship, view, and other shapes and drop them onto the drawing page to create your model.  
For details about how to create a database from scratch, see [“Creating a database model from scratch”](#) in Chapter 7, “Creating database models.”

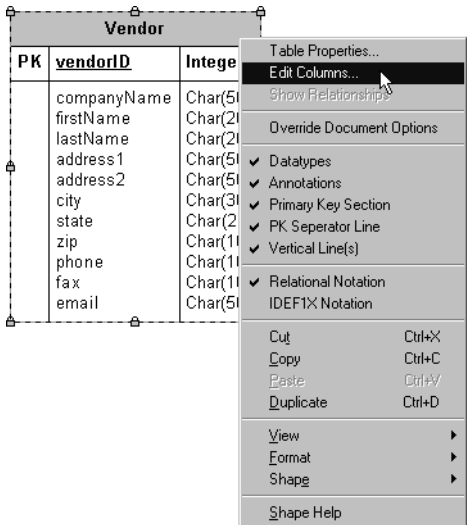
# Refining a database model

After you have a database model started, you can use the tools in Visio Enterprise to refine it. You can add, modify, or delete the tables, columns, and relationships in a model, and you can choose to display more or less detail about your database in the model diagram. You can also define and edit advanced settings such as data types, stored procedures, and extended attributes. Finally, you can validate your model to ensure that it is free of errors.

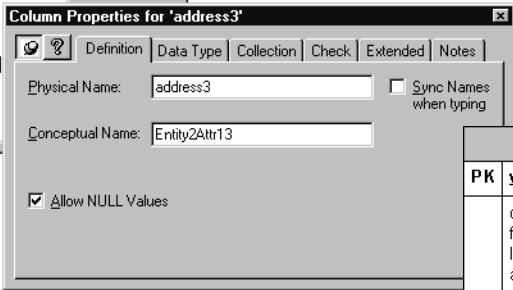
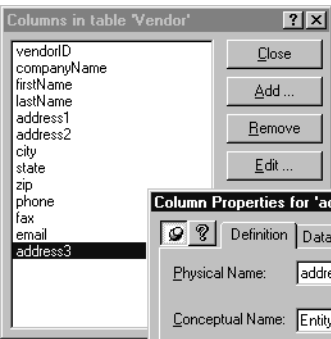
**Figure 6-7** Adding a column to an existing table

When you use Visio Enterprise to refine a database model, you can modify the elements that make up the database. In this example, a new column is added to an existing table.

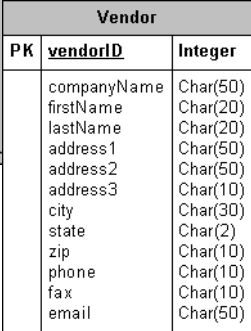
**1** Right-click a table, then choose Edit Columns.



**2** Click Add, then Edit.



**3** Then, customize the column's properties.





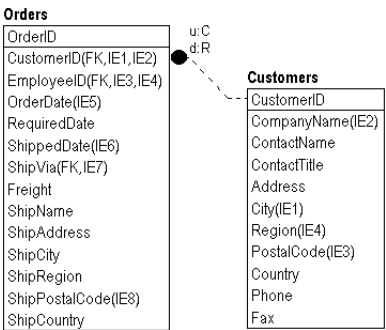
## Customizing a model's on-screen appearance

The way the shapes in a model appear on the screen is determined by the notation type you select for the Symbol Set and other options you select in the Database Document Options dialog box. Visio Enterprise supports IDEF1X and Relational notation. You can also choose to display relationships for either notation using Crow's Foot symbols.

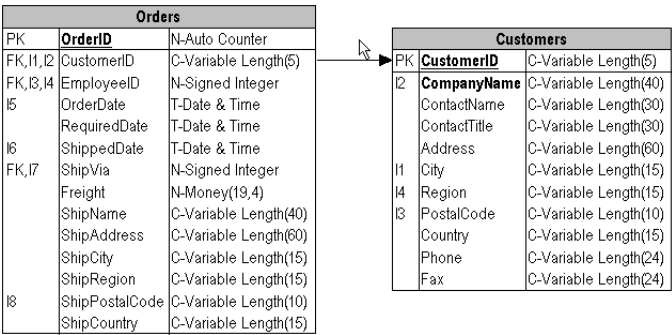
In addition to choosing the notation your database model uses, you also determine the way Visio Enterprise draws relationships and the amount of detail shown in your model. For example, you can choose whether to display referential integrity, all or only selected columns in a table, whether data types are visible, and many other options.

**Figure 6-8**

Changing notation options changes the appearance of a model on the drawing page, but does not affect the information in the model. Other options determine the level of detail visible in the diagram.



**IDEF1X notation with referential integrity notation visible**



**Relational notation with data types visible**

If you right-click a table, view, or type, then choose **Override Document Options**, you can customize display options for that element.

### To choose display options:

- 1 Choose Database > Document Options.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the General tab in the Database Document Options dialog box, select a Symbol Set option. On the Table and Relationship tabs, select options to display or hide information about the model.

For more details about working in this dialog box, see [“Setting display options for a database model”](#) in Chapter 7, “Creating database models.”

## Using right-click shortcut menus

You can right-click any component of your model to display a shortcut menu of additional commands for refining it. For example, you can right-click a table to access the Edit Columns command, which you can use to edit a column’s properties. Or, you can right-click a relationship to access the Relationship Properties dialog box, where you can edit foreign key definitions and change referential integrity settings.

## Using the Code window, Tables & Views window, and Types window

You can further refine a database model using the Visio Enterprise Code Window, Tables & Views window, and Types window. To open any of these windows, choose Database > View, then choose the window you want to display. Following are descriptions of each window:

**Code window** Lists all code for a particular database platform that is associated with your model, including code extracted from a database during reverse engineering. The Code window also provides access to the Code Editor, where you can revise existing code or write new code. For details about using the Code window and the Code Editor, see [Chapter 9, “Reverse engineering database code.”](#)

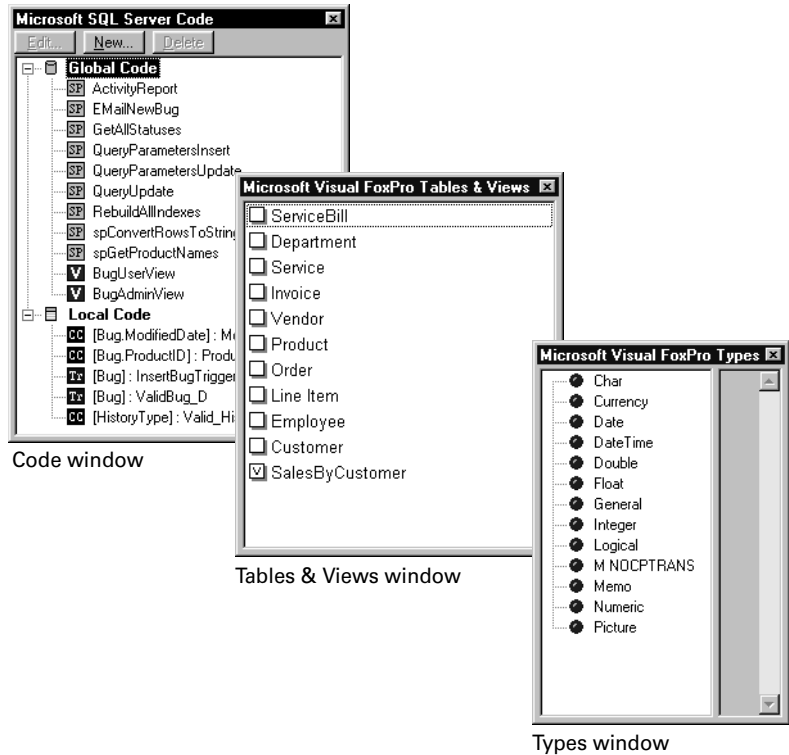
**Tables & Views window** Appears automatically when you use the Reverse Engineer Wizard, and lists the names of tables and views extracted from the database. When you drag elements from the Tables & Views window and drop them into your diagram, they appear in your model, where you can view them along with any associated relationships.

**Types window** Contains the built-in physical data types for your target database, portable data types you created using the User Defined Data Types dialog box, and composite types you’ve defined. For details about user-defined types,

see “Working with data types” in Chapter 8, “Refining database models.” Once you are familiar with the physical data type requirements of your DBMS, you can assign data types to specific columns in the model.

**Figure 6-9**

The Code window, Tables & Views window, and Types window provide advanced options for refining a database model.



## Validating a model

As you refine your database, you can validate it periodically to ensure that it is free of errors. Visio Enterprise performs two types of validation.

- **Logical validation** checks the database model to ensure that it conforms to generally accepted database design standards. For example, circular table relationships are identified when you validate a model, along with duplicate column names.
- **Physical validation** checks the model to be sure it is valid for the target DBMS. For example, Microsoft SQL Server allows only one column in a table to have an identity property set. A table containing more than one column with this physical attribute will not generate an error during logical validation, but will fail physical validation.

Validating your model as you refine it can help ensure that your database is as efficient as possible, and can streamline the process of using the Generate, Update Model, and Update Database Wizards.

**To validate your model:**

- 1 Choose Database > Model > Check.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 In the Output window, double-click any error or warning to highlight the source of the problem in the model.
- 3 Correct the cause of the error or warning.
- 4 Repeat steps 2 and 3 for each error or warning in the Output window.

For details about checking for and correcting errors, see [“Validating a database model”](#) in [Chapter 8](#), [“Refining database models.”](#)

## Connecting to a database

Visio Enterprise provides robust functionality for generating and updating databases from your database model. Using its own database drivers, which are optimized to provide DBMS-specific functionality, in association with your generic or vendor-supplied ODBC (Open Database Connectivity) drivers, Visio Enterprise makes it possible for you to map your database model to any supported DBMS.

For a list of supported DBMSs and for details about configuring drivers and connecting to your DBMS, see [Chapter 10](#), [“Connecting to a DBMS.”](#)

## Generating a database

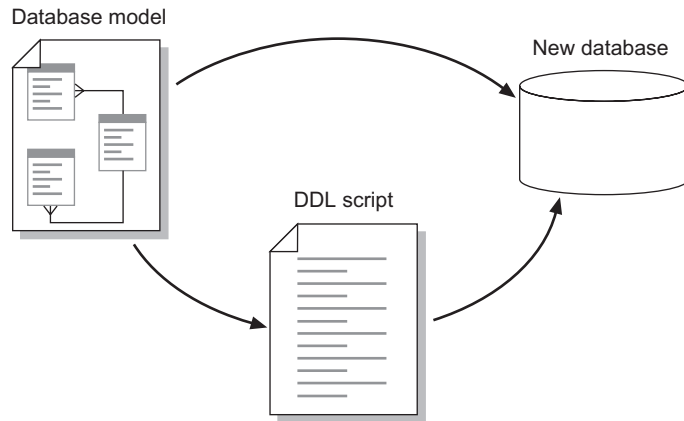
When your model is complete and you’re ready to generate a new database from it, you can use the Generate Wizard to create either a new database schema directly (if you are working with a DBMS that supports this functionality) or to create an editable DDL script that can be used by your DBMS to create the database schema. You can also choose to create both simultaneously.

The Generate Wizard assists you in choosing generate options, selecting drivers, connecting to a data source, and previewing changes. As a final step, the Generate Wizard validates your model to ensure that the schema information in the model does not violate any restrictions imposed by the target DBMS.

If your model is free of critical errors, Visio Enterprise generates a new database, DDL script, or both for you. If there are errors or warnings, Visio Enterprise lists them in both the wizard and the Output window, so you can correct them in the database model.

**Figure 6-10**

From your model, you can generate a new database directly, an editable DDL script, or both.



**To generate a database schema:**

- 1 Choose Database > Generate to open the Generate Wizard.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 Follow the instructions in the wizard.

For details about generating database schemas and using the Generate Wizard, see [Chapter 11, “Generating a database schema.”](#)

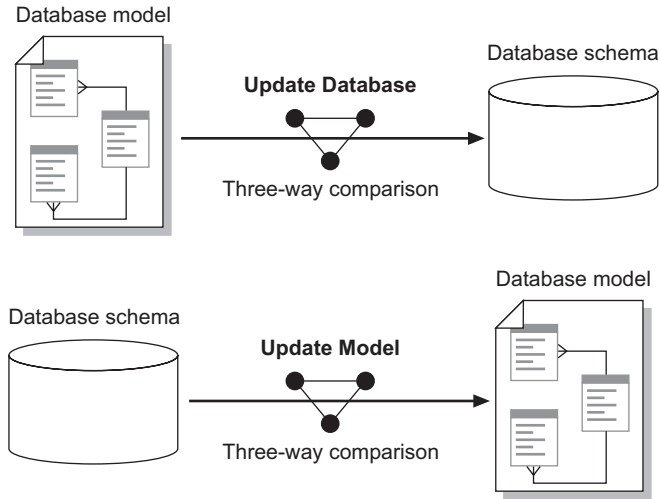
## Updating a model or a database

Whether you have created a new logical model or are reverse engineering an existing database, Visio Enterprise can synchronize any changes made to the database while you refine your model. Using a unique three-way comparison that tracks differences between your model and the database, Visio Enterprise makes it possible for you to migrate changes from the model to the database, or from the database to the model.

To update a model from the database, use the Update Model Wizard, which provides options for resolving discrepancies. To update a database from a model, typically the final step in the reverse engineering process and also called forward engineering, use the Update Database Wizard.

**Figure 6-11**

Using a three-way comparison process, Visio Enterprise can update a database from your model or update your model from a database.



**To update a model:**

- 1 Choose Database > Model > Update to open the Update Model Wizard.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 Follow the instructions in the wizard.

For details about synchronizing your logical model with your database and using the Update Model Wizard, see [Chapter 12, “Updating models and databases.”](#)

**To update a database:**

- 1 Choose Database > Update to open the Update Database Wizard.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 Follow the instructions in the wizard.

For details about migrating changes from your model to your database and using the Update Database Wizard, see [Chapter 12, “Updating models and databases.”](#)



### To create a report:

- 1 Choose Database > Report to open the Report Wizard.  
**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.
- 2 Select the type of report you want to create, then click Finish.
- 3 In the Report window, define and format the information you want to include in the report.
- 4 Click Print, Preview, or Export To RTF to generate the report.  
For details about creating reports and using the Report Wizard, see [Chapter 13, “Creating reports.”](#)

## Next steps

This chapter has introduced the tools you use and the basic process you follow to create and work with database models in Visio Enterprise. To explore the Visio Enterprise database modeling solution, see the topics below.

<i>If you want to</i>	<i>See</i>
Create a new logical database model	<a href="#">Chapter 7, “Creating database models”</a>
Refine a logical model	<a href="#">Chapter 8, “Refining database models”</a>
Reverse engineer code	<a href="#">Chapter 9, “Reverse engineering database code”</a>
Connect to a database or understand how Visio Enterprise uses drivers and data sources	<a href="#">Chapter 10, “Connecting to a DBMS”</a>
Generate a new database	<a href="#">Chapter 11, “Generating a database schema”</a>
Update your logical model with changes from the physical database	<a href="#">Chapter 12, “Updating models and databases”</a>
Update your database based on your logical model	<a href="#">Chapter 12, “Updating models and databases”</a>
Create a report	<a href="#">Chapter 13, “Creating reports”</a>



# Creating database models

Visio® Enterprise supports the goals of data modeling: to capture real-world information about an application domain accurately, and in a way that is meaningful and understandable to both the user and the database designer. With the Database Model solution in Visio Enterprise, you can design a database schema and re-engineer existing databases by creating a **model**. A model is a Visio Enterprise diagram in which you represent the logical structure of a database using shapes, which you can then modify to refine the database schema. Ultimately you can forward engineer your model to an existing database or use it to generate a new database schema.

This chapter describes how to start working in the Database Model solution to create a new database model from scratch or from an existing database or model created in another program. For details about working with shapes to refine your model, see [Chapter 8, “Refining database models.”](#)

## Topics in this chapter

- Creating a new database model ..... 130
- Creating a database model from scratch ..... 133
- Reverse engineering an existing database ..... 135
- Importing a data model ..... 141
- Customizing your database model diagram ..... 142
- Next steps ..... 144

# Creating a new database model

Whether you start from scratch or from an existing database or model, you begin a new database model in Visio Enterprise by opening the Database Model solution. This solution includes stencils of shapes that represent the components of relational and object-relational data models. In addition, opening this solution adds the Database menu to the Visio Enterprise toolbar, which contains the commands you use to work with database model diagrams.

## To open the Database Model solution:

- Choose File > New > Database > Database Model.

Visio Enterprise creates a new, empty drawing page and opens the Entity Relationship and Object Relational stencils.

After you open the Database Model solution, you can start to create your database model in three ways. You can

- Drop the Entity Relationship or Object Relational shapes on the drawing page to construct a model from scratch, which you can later use to generate a new database schema.
- Reverse engineer all or part of the schema of an existing database, which automatically creates a model on the drawing page.
- Import a data model created with Visio® Professional, VisioModeler™ (formerly InfoModeler®), or PLATINUM ERwin.

## Representing relational and object-relational databases

In the Database Model solution, you can create models that represent either relational or object-relational databases. The Database Model solution opens with the Entity Relationship stencil for creating relational database models, and the Object Relational stencil for creating object-relational database models with extended functionality such as composite data types, type inheritance, and table inheritance. Each type of model in Visio Enterprise has its own set of tools and professional reports that allow you to focus on your task of clearly and accurately describing the information requirements of users.

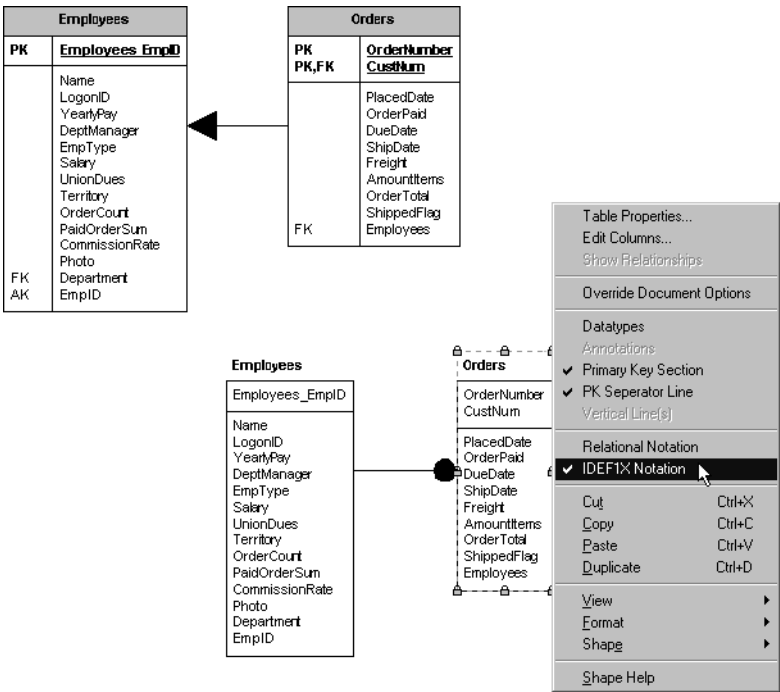
By working with the shapes from one stencil, you can choose the modeling method you want to use to build your database model diagram. Relational and object-relational notations share many common elements.

- **Tables or entities**, which you can create using the Entity shape.
- **Columns or attributes**, which you can create by right-clicking an Entity shape, then choosing Edit Columns from the shortcut menu. Notation for primary keys, foreign keys, alternate keys, and indexes is supported.
- **Relationships**, which you can create using the Relationship shape, which includes support for referential integrity and IDEF1X cardinality constraints.

You use the shapes on the Entity Relationship or Object Relational stencil to represent tables, columns, and other components as they appear in a database.

For details about these and other ways of customizing a database model, see “Customizing your database model diagram” later in this chapter. For details about each of the Entity, Relationship, and other shapes you work with, see the specific sections in Chapter 8, “Refining database models.”

**Figure 7-1**  
Visio Enterprise gives you the flexibility to switch between modeling notations easily without losing any information in the model.



## Setting display options for a database model

With either stencil, you can choose the notation symbolized in the shapes.

- **IDEF1X** Describes a data model as entities with attributes that participate in relationships. Entities are depicted as rectangles with the entity's name on top and its attributes inside. A relationship between entities is portrayed by a 1-D connector shape.
- **Relational** Uses tables, columns, and keys to describe your data model. Tables are represented as rectangles with the table's name across the top. Columns of the table are listed vertically underneath the name of the table. A foreign key relationship is shown by a 1-D connector shape.

As you work, you can switch easily between notations—helpful if you are collaborating with a colleague who prefers a different notation. In addition, for either notation, you can further specify whether to show relationships using Crow's Foot notation, a variation of connector style that indicates relationships between entity shapes.

You can also specify the level of detail displayed in a database model diagram. By showing and hiding levels of detail, you can create a more detailed or condensed view of your model. For example, you can display just the elements you want, such as the table name, primary keys, foreign keys, and relationships, or you can choose to look only at primary keys and foreign keys and not to display data types.

### To customize how a database model appears:

- 1 Open the Visio Enterprise Database Model file that represents your model.
- 2 Choose Database > Document Options.
- 3 On the tabs in the Database Document Options dialog box, do the following:

**General tab** Choose the graphical notation set and name options you want to display.

**Table tab** Choose column, table, view, and type display attributes, primary key order, and data type display, and whether to display IDEF1X optionality.

**Relationship tab** Choose whether to display relationships. If you choose to display relationships, select whether to represent them with Crow's Foot notation, whether to display cardinality and referential integrity, and how to display the relationship verb phrase and name.

*Optional* To keep the settings you choose as the default settings, click the Defaults button, then choose Set As. To return the settings to the most recently saved defaults, choose Restore. To return to the Visio Enterprise defaults, choose Restore Original.

**4** Click OK.

**TIP** For an individual shape in your model, you can override the notation preferences set for the model. For example, if your client has a specialized notation for tables, you can override the default settings of the model to show a table with different notation. To do this, right-click the shape you want to change, choose Override Document Options from the shortcut menu, then manually set the properties you want.

## Creating a database model from scratch

When you design a model from scratch, you create a new model diagram with an empty drawing page, and then use the shapes on the Entity Relationship or Object Relational stencil to design a database schema.

In general, you follow these steps when constructing a data model.

- 1** Draw a table (or entity).
- 2** Insert columns (or attributes).
- 3** Draw the relationships between the tables (or entities).
- 4** Define and set the properties for each component of your data model.

Once you have completed these basic steps, you refine your design, repeating the general procedure until you're ready to use your design to generate a schema for your target DBMS (database management system).

The following procedure outlines the general steps of starting a new model diagram and adding shapes to it. Details about working with IDEF1X and Relational notation, and refining the definition of each model component, are described later in this chapter, as noted in the procedure.

### To create a new database model diagram from scratch:

- 1 Choose File > New > Database > Database Model.

Visio Enterprise creates a new, empty drawing page and opens the Entity Relationship and Object Relational stencils.

- 2 Choose Database > Document Options, then on the General tab, choose the notation you want to use: IDEF1X or Relational. Choose other table and relationship options, then click OK.

The notation affects the appearance of shapes on the drawing page, but does not change their appearance or name on the Entity Relationship and Object Relational stencils.

- 3 Drag an Entity shape from a stencil and drop it onto the drawing page.
- 4 To add attributes or columns, right-click the Entity shape, then choose Edit Columns from the shortcut menu. Click Add to create a new column, select it from the list, then click Edit to define its properties. When you are finished, click the Close box to close the Edit Columns dialog box.

For details, see [“Working with columns”](#) in [Chapter 8, “Refining database models.”](#)

- 5 Right-click the Entity shape, then choose Table Properties from the shortcut menu. Define the table and its properties, such as primary ID and indexes, on the tabs in this dialog box, then click OK.

The appearance of the table shape in the drawing reflects its definition and the notation you selected.

For details about setting table properties, see [“Working with tables”](#) in [Chapter 8, “Refining database models.”](#)

- 6 Add Entity shapes to the drawing page and define attributes or columns for the other entities or tables you want in the model.
- 7 Drag a Relationship shape onto the drawing page. Hold the Ctrl key while you drag an endpoint to one Entity shape, then release the mouse. Repeat for the other endpoint of the Relationship shape.

The endpoints turn red when the shapes are glued.

- 8 Right-click the relationship, then choose Relationship Properties from the shortcut menu. Define the relationship and its properties, such as referential integrity and cardinalities, on the tabs in this dialog box, then click OK.

The appearance of the Relationship shape in the drawing reflects its definition and the notation options you have selected. For example, an identifying relationship is indicated by a solid line and a child is indicated by a black dot.

**NOTE** After you define a relationship, Visio Enterprise automatically inserts the primary key of the parent table into the child table as its foreign key.

For details, see [“Working with relationships”](#) in Chapter 8, [“Refining database models.”](#)

- 9 Define relationships between the remaining Entity shapes on the drawing page.

You can add views, categories, code, and further refinements to your model. For details, see [“Using categories”](#) in Chapter 8, [“Refining database models”](#) and [“Adding code to a model”](#) in Chapter 9, [“Reverse engineering database code.”](#)

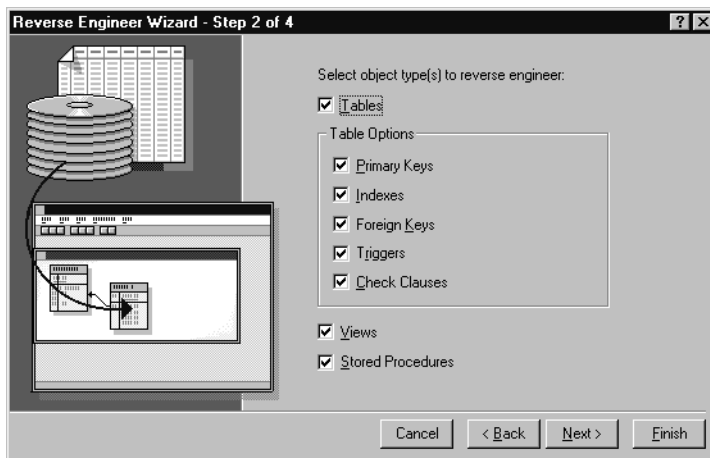
## Reverse engineering an existing database

Probably the simplest and most powerful way to create a new database model in Visio Enterprise is to reverse engineer an existing database. The Reverse Engineer Wizard in the Database Model solution automates the process of **extracting** the schema from a database that you specify, including Microsoft Repository. You use the wizard to specify the platform-specific database objects that you want to reverse engineer: tables, primary keys, indexes, foreign keys, triggers, check clauses, views, and stored procedures. Then the wizard extracts, analyzes, and validates the schema into a database model diagram that you can review and modify.

The Reverse Engineer Wizard steps you through the process of connecting to the target DBMS that you want to reverse engineer. To create a connection, you can use the wizard to specify an installed Visio Enterprise driver and the 32-bit ODBC (Open Database Connectivity) data source. However, for best results, set your default driver to the target database you want to reverse engineer before running the wizard. The information that the wizard can extract depends upon the capabilities of the DBMS. If your DBMS makes some of the schema definition available, Visio Enterprise makes those options available in the Reverse Engineer Wizard, as Figure 7-2 shows, and dims the other options.

**Figure 7-2**

You can choose the schema information to extract in the Reverse Engineer Wizard.



The wizard notifies you of its progress in the Output window. When the extraction process is complete, the wizard adds the extracted structures to the currently open model diagram and displays them in the Tables & Views window.

## Connecting to a data source

The first time you extract information from a database, you must specify how to locate and communicate with that database by associating the appropriate drivers. Visio Enterprise uses one of its own database drivers, in conjunction with a 32-bit ODBC driver, to connect to and extract information from a database.

You can set up the drivers before you run the Reverse Engineer Wizard by choosing Database > Drivers. Or you can use screens in the wizard to perform this step. Either way, you must

- Select an installed Visio Enterprise database driver.
- Associate a 32-bit ODBC driver with an installed Visio Enterprise database driver.
- Select or create a 32-bit ODBC data source. For details about creating a data source, see the documentation for the Microsoft 32-bit ODBC Data Source Administrator.

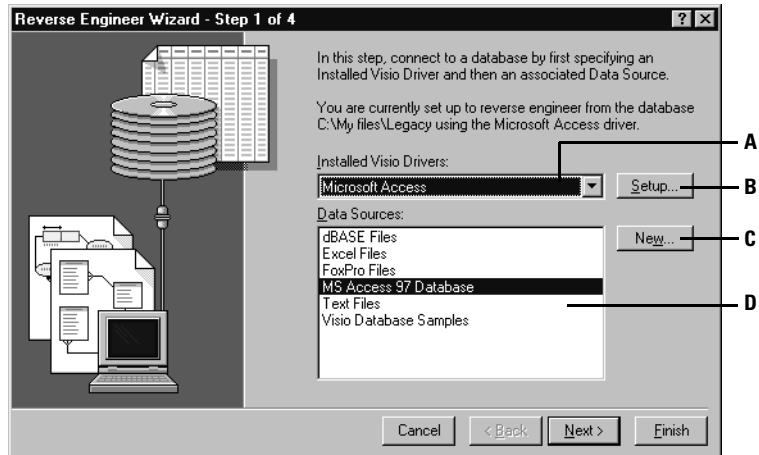


Once you have associated drivers, either in the Database Drivers dialog box or the Reverse Engineer Wizard, you do not need to repeat this step unless you want to change an association you have made.

**Figure 7-3**

Connecting and associating database drivers

- A** Select an installed Visio Enterprise driver.
- B** Click to associate a Visio Enterprise driver with a 32-bit ODBC driver.
- C** Click to create a new data source.
- D** Select the data source that points to your database.



The drivers you installed with Visio Enterprise appear in the Installed Visio Drivers box in the Reverse Engineer Wizard. After you choose the Visio Enterprise driver you want to use, you can associate it with the 32-bit ODBC driver for your target database if you have not already done so. The 32-bit ODBC driver that you choose determines the data sources you see in the list of available data sources. If you do not create an association, Visio Enterprise lists all the data sources on your system. The association tells Visio Enterprise which 32-bit ODBC driver to use if you have more than one installed on your system.

When you're starting a new model, it's probably easier to set up the driver associations before running the Reverse Engineer Wizard. For details, see [Chapter 10, "Connecting to a DBMS."](#)

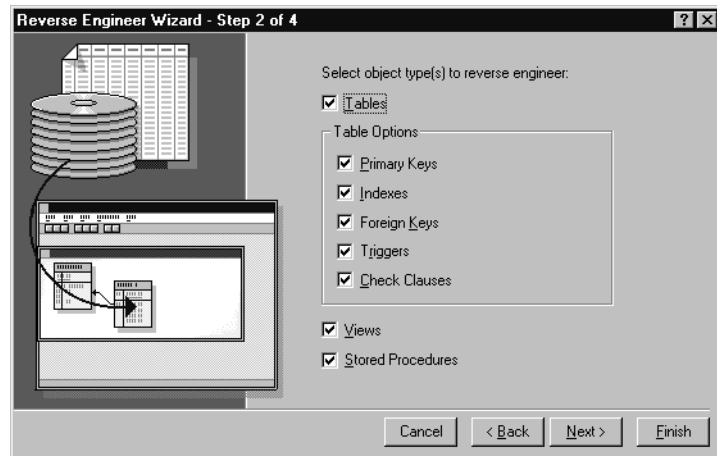
## Reverse engineering code

A database can include SQL code in the form of stored procedures, triggers, check clauses, functions, and so on. This code is preserved when you reverse engineer a database, so that you can work with code components of your model in Visio Enterprise. When you reverse engineer a database that contains code, the Reverse Engineer wizard copies the SQL code as text and then adds the text to the model as one of the following types of code:

- Stored procedure
- Trigger
- Function
- Check clause
- View definition (but not the join criteria for views)
- Raw DDL (Data Definition Language code; that is, other SQL code that does not fall into one of the above categories)

**Figure 7-4**

In the Reverse Engineer Wizard, you can choose the types of code to extract with the database schema.



After you reverse engineer a database that contains code, Visio Enterprise lists each code component in the Code window in the order it was extracted. To display the Code window, choose Database > View > Code Selector. For details about working in the Code window, see [“Viewing and editing code”](#) in Chapter 9, [“Reverse engineering database code.”](#)

The Code window displays the version of code specific to the platform from which it was reverse engineered. However, you can transfer code to a different platform to store multiple versions of code with your model. For details, see [“Transferring code to a new target platform”](#) in Chapter 9, [“Reverse engineering database code.”](#)

## Extracting a schema using the Reverse Engineer Wizard

You can extract all or part of the schema of an existing database into a new or existing model that you can then refine. The Reverse Engineer Wizard connects to your target DBMS, extracts the schema information you choose, and adds it to the currently open model diagram.

**NOTE** For best results, before you run the Reverse Engineer Wizard, choose Database > Drivers to set the default driver to the target back end. This step ensures that the wizard will map the native data types correctly and that all the code extracted by the wizard will be correctly displayed in the Code window.

### To extract a database schema into a model diagram:

- 1 Choose File > New > Database > Database Model to start a new model. Or open an existing one created with the Database Model template.
- 2 Choose Database > Reverse Engineer to begin working in the Reverse Engineer Wizard.
- 3 On the first screen, do the following:

From the Installed Visio Drivers list, choose the driver. If you have not already associated a Visio Enterprise enhanced driver with a particular 32-bit ODBC data source, click Setup to create the association now.

Under Data Sources, choose the source of the database you're updating, then click Next. If you have not already created a data source for the existing database, click New, then follow the instructions in the wizard. For details, see [“Creating and setting up an ODBC data source”](#) in Chapter 10, “Connecting to a DBMS.”

When you create a new source, Visio Enterprise adds its name to the Data Sources list.
- 4 In the Connect Data Source dialog box, type a user name and password, then click OK. In the Select Database dialog box, choose the database you want to reverse engineer, then click OK.
- 5 Check the options for the type of information you want to extract, then click Next.
- 6 Check the tables (and views, if any) that you want to extract, or click Select All to extract them all, then click Next.
- 7 If you checked Stored Procedures in step 5, on the next screen check the procedures that you want to extract, or click Select All to extract them all, then click Next.

- 8 Verify that you are extracting the information you want, then click Finish.  
The wizard extracts the selected information and displays notes about the extraction process in the Output window.
- 9 Drag the tables you want to view from the Tables & Views window onto the drawing page.  
For details about working with extracted code, see [“Viewing and editing code”](#) in [Chapter 9, “Reverse engineering database code.”](#)  
For details about working in your model, see [Chapter 8, “Refining database models.”](#)

**TIP** As you work in your database model diagram, keep it in sync with the database from which it was extracted by choosing Database > Model > Update. For details, see [“Running the Update Database Wizard”](#) in [Chapter 12, “Updating models and databases.”](#)

## Extracting a database schema from Microsoft Repository

Using Microsoft Repository you can share database models with colleagues and use your models with other Microsoft Repository–enabled data modeling tools. Visio Enterprise supports storage and retrieval of your database model in Microsoft Repository using the Database Information Model (DBM) open format. Information about your database model is stored in Microsoft Repository in either a SQL Server or Microsoft Jet database, which is independent of the target implementation platform for your database.

Before you can extract a database schema from Microsoft Repository, you must establish a connection with Microsoft Repository. You do this using the Microsoft DBM Repository driver provided with Visio Enterprise. For details about connecting to Microsoft Repository, see [“Connecting to Microsoft Repository”](#) in [Chapter 10, “Connecting to a DBMS.”](#)

### To extract a database model from Microsoft Repository:

- 1 In Visio Enterprise, choose File > New > Database > Database Model, then choose Database > Reverse Engineer.
- 2 In the Reverse Engineer Wizard that appears, choose Microsoft DBM Repository from the list of available drivers, then click Next.
- 3 Establish a connection with Microsoft Repository.  
For details about connecting to Microsoft Repository, see [“To connect a Visio Enterprise model to Microsoft Repository:”](#) on page 198.

- 4 In the Select Schema To Extract dialog box that appears, choose the schema you want to work with, then click OK.
- 5 Check the options for the type of information you want to extract, then click Next.
- 6 Check the tables (and views, if any) that you want to extract, or click Select All to extract them all, then click Next.
- 7 If you checked Stored Procedures as an option in step 5, on this screen check the procedures that you want to extract, or click Select All to extract them all, then click Next.
- 8 Verify that you are extracting the information you want, then click Finish. The wizard extracts the selected information and displays notes about the extraction process in the Output window.
- 9 Drag the tables you want to view from the Tables & Views window onto the drawing page.

For details about working with extracted code, see [“Viewing and editing code” in Chapter 9, “Reverse engineering database code.”](#)

For details about working in your model, see [Chapter 8, “Refining database models.”](#)

**TIP** As you work in your database model diagram, keep it in sync with the database from which it was extracted by choosing Database > Model > Update. For details about updating, see [“Running the Update Model Wizard” in Chapter 12, “Updating models and databases.”](#)

## Importing a data model

If you have an existing data model created in Visio Professional, VisioModeler, or PLATINUM ERwin, you can import it as a physical model in Visio Enterprise. When you work in the Database Model solution, you can import data models in the following formats:

- Visio Professional 5.0 entity relationship or Crow’s Foot diagram (.vsd) files.
- VisioModeler (formerly InfoModeler) 1.5, 2.0, 3.0, and 3.1 dictionary model (.imd) files.
- PLATINUM ERwin 2.6 and 3.0 (.erx) files.

After you import an existing model, Visio Enterprise displays all its tables and views in the Tables & Views window. You can then choose the tables to display on the drawing page, set notation options, and work with the imported model as you would any other Visio Enterprise database model diagram.

**To import a model file from another program:**

- 1** Choose File > New > Database > Database Model to open a new Visio Enterprise database model document.  
Or open an existing Visio Enterprise database model document.
- 2** Choose Database > Import/Export > Import <model type>.  
Choose the Import ERwin ERX, Import Visio Database Diagram, or Import VisioModeler IMD Model command.
- 3** Type the path and file name for the model you want to import. Or click the Browse button to locate the model file, then click Open.
- 4** Click OK in the Import dialog box.  
Visio Enterprise imports the file and displays its progress in the Output window. The imported tables are displayed in the Tables & Views window.
- 5** In the Tables & Views window, select the tables you want to model, then drag them onto the drawing page.

## Customizing your database model diagram

You can customize the Visio Enterprise environment while you work in the Database Model solution by customizing the behavior of the shapes that represent tables, columns, and relationships in your database model. For example, you can

- Set the behavior when you remove a table or foreign key in a database model diagram.
- Choose how foreign keys are propagated.
- Determine how name conflicts are resolved and how default name prefixes are created for the elements in a database model.

To customize these preferences, choose Database > Modeling Preferences. The preferences you set remain in effect across all Visio Enterprise sessions until you choose different settings.

### **To set behavior options for a database model diagram:**

- 1** Open the Database Model file that represents your model.
- 2** Choose Database > Modeling Preferences.  
The Database Modeling Preferences dialog box opens.
- 3** On the Logical Diagram tab in the Database Modeling Preferences dialog box, choose the options you want to set from the following:
  - When Removing An Object From The Diagram** Specifies whether to remove a deleted object only from the drawing page, retaining it in the Tables & Views window, or remove it from the drawing file (and model) altogether.
  - Show Relationships After Adding Table To Diagram** Shows relationships on the current page between a newly added table and any other tables on the page.
  - Show Relationships After Adding Type To Diagram** Shows relationships between a newly added type in the diagram and any other types on the page.
  - Sync Conceptual And Physical Names In New Tables And Columns When Typing** Changes how the Table Properties and Column Properties dialog boxes work so that typing a name in one field automatically creates the name in the other field.
- 4** Click the Logical Misc tab, then choose the options you want to set from the following:
  - FK Propagation** Specifies the behavior when adding or deleting foreign keys in a model or when Visio Enterprise encounters a name conflict when a foreign key is added.
  - Default Name Prefixes** Specifies the prefix that is added to the suffix under Default Name Suffixes to form the default name for objects added to a model.
  - Default Name Suffixes** Specifies the suffix that is added to the prefix under Default Name Prefixes to form the default name for objects added to a model.
  - FK Name Generation Option** Specifies how to construct the default foreign key name that appears in a model.
- 5** When you are finished, click OK.

# Next steps

Typically, your next task is to update and refine your database model using the tools in Visio Enterprise. The following topics provide the information you need.

<i>If you want to</i>	<i>See</i>
Refine your model	<a href="#">Chapter 8, “Refining database models”</a>
Synchronize the model with the database you reverse engineered	<a href="#">Chapter 12, “Updating models and databases”</a>
Print or preview a report of your model	<a href="#">Chapter 13, “Creating reports”</a>



# Refining database models

Visio® Enterprise offers sophisticated functionality in the Database Model solution for working with tables, columns, views, and relationships. As you refine a database model in Visio Enterprise, you can define and edit settings such as data types, referential integrity, indexes, and extended attributes. After you refine your database model, you can check it for both logical and physical validity, and then generate a new or update an existing database.

This chapter explains how to use the advanced features in Visio Enterprise to refine database models. This chapter assumes that you are familiar with database concepts and architecture.

## Topics in this chapter

➤ Working with tables.....	146
➤ Working with columns .....	152
➤ Working with data types .....	154
➤ Working with views .....	161
➤ Working with relationships.....	164
➤ Creating and editing indexes.....	168
➤ Setting extended attributes .....	171
➤ Validating a database model .....	173
➤ Next steps.....	174

# Working with tables

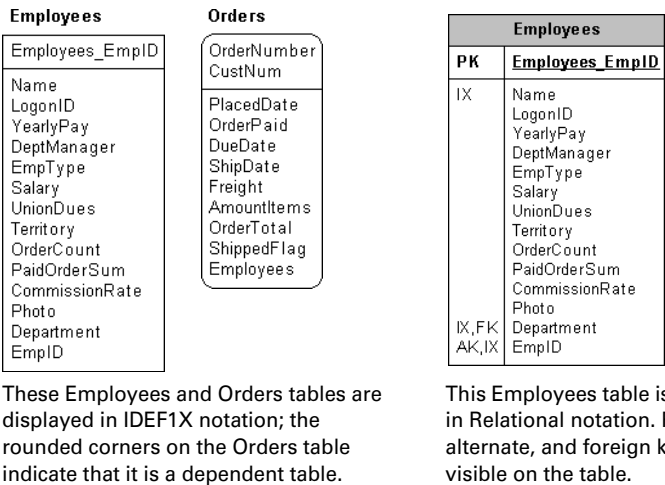
With Visio Enterprise, you can add new tables to and remove existing tables from the database model, and you can edit table properties. All of your changes are reflected in the database schema when you generate a new database or update an existing one. For details about generating a new database, see [Chapter 11, “Generating a database schema.”](#) For details about updating an existing database, see [Chapter 12, “Updating models and databases.”](#)

The way tables are displayed in Visio Enterprise depends on whether you choose to display the model in IDEF1X notation or Relational notation, the level of detail you want to display, and whether you choose to display relationships using Crow’s Foot notation.

Tables displayed in IDEF1X notation appear as rectangles with the table’s conceptual name above. Tables displayed in Relational notation have a shaded box at the top of the table containing the table’s conceptual name.

IDEF1X tables are classified as **independent** or **dependent**; independent tables include no foreign keys as primary keys and are drawn as rectangles with square corners, while dependent tables include one or more foreign keys as primary keys and are drawn as rectangles with rounded corners. There is no graphical notation for independent and dependent tables in models displayed using Relational notation.

**Figure 8-1**  
Tables displayed in IDEF1X notation and Relational notation



**To add a new table to the database model:**

- 1 Open the Visio Enterprise Database Model file that represents your model.
- 2 From either the Entity Relationship or Object Relational stencil, drag an Entity shape onto the model drawing page.

Visio Enterprise adds a table to your model. You can set the table attributes, add columns, and specify relationships with other tables in the model. For details, see [“Working with columns”](#) and [“Working with relationships”](#) later in this chapter.

**To remove a table from the database model:**

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Select the table you want to remove, then press Delete.
- 3 In the Delete Object dialog box, choose Yes if you want to remove the table from the model and the Tables & Views window, or No to remove it from the drawing page but not from the model or the Tables & Views window.

**NOTE** If you changed the default modeling preferences for removing objects from the diagram to an option other than Ask User What To Do, the Delete Object dialog box will not appear. To check your settings for object removal, choose Database > Modeling Preferences, then click the Logical Diagram tab. For details about setting modeling preferences, see [“Customizing your database model diagram”](#) in Chapter 7, “Creating database models.”

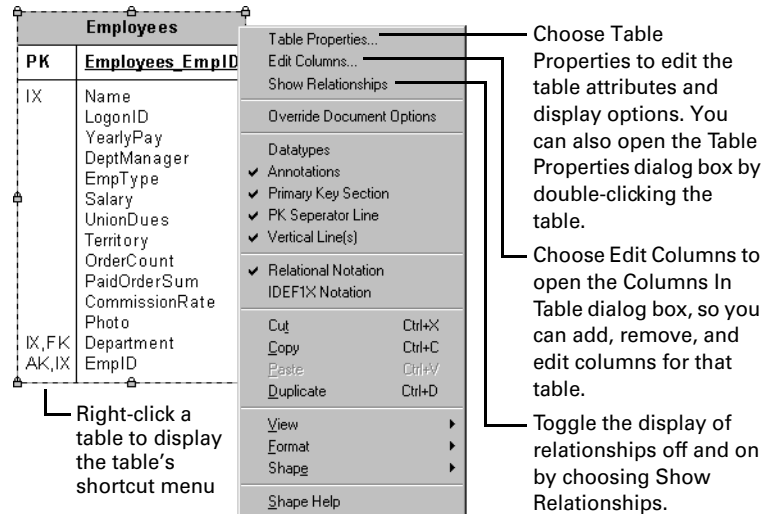
### To edit the properties of a table in the database model:

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Right-click the table whose properties you want to edit, then choose Table Properties from the shortcut menu.

**TIP** You can also double-click the table shape to display the Table Properties dialog box.

**Figure 8-2**

Commands for customizing your table



- 3 In the Table Properties dialog box, click the tab that contains the information you want to edit, and make your changes. The database model diagram updates automatically. You can choose Edit > Undo to reverse the changes. Following is a brief description of each tab and the options it contains.

**Definition tab** Specify physical and conceptual names for the table and choose whether to synchronize the names. You can specify a name space value for the table to distinguish it from similarly named tables in the model. This tab also displays information reported by the host database management system (DBMS) about the database owner and the path of the source database. The defining type field is available only when a table is empty, and lists all composite data types so you can create a typed table.

**Primary ID tab** Edit, define, or delete primary keys from a list of available columns and choose whether to create an index on primary keys.

**Indexes tab** Create, edit, define, rename, delete, or set extended attributes for indexes. Use this tab to specify the type of index you create for a particular column. For details about indexes and extended attributes, see [“Creating and editing indexes”](#) and [“Setting extended attributes”](#) later in this chapter.

**Triggers tab** Add, edit, remove, or preview the code for triggers that are included in the DDL (Data Definition Language) script created when you use the Generate Wizard or Update Database Wizard. For details, see [“Creating trigger code”](#) in Chapter 9, [“Reverse engineering database code.”](#)

**Check tab** Add, edit, remove, or preview the code for check clauses that are included in the DDL script created when you use the Generate Wizard or Update Database Wizard. For details, see [“Creating check clauses”](#) in Chapter 9, [“Reverse engineering database code.”](#)

**Extended tab** Set DBMS-specific extended attributes for use with the Visio Enterprise database drivers.

**Notes tab** Type text to document the table. Text entered here appears in the Verbalizer window.



**NOTE** To get online help for an option on a tab, click the question mark button in the dialog box, then click the option for which you want help. Or right-click the option, then choose What’s This.

- 4 Click the Close button or press Alt+F4 to close the dialog box.



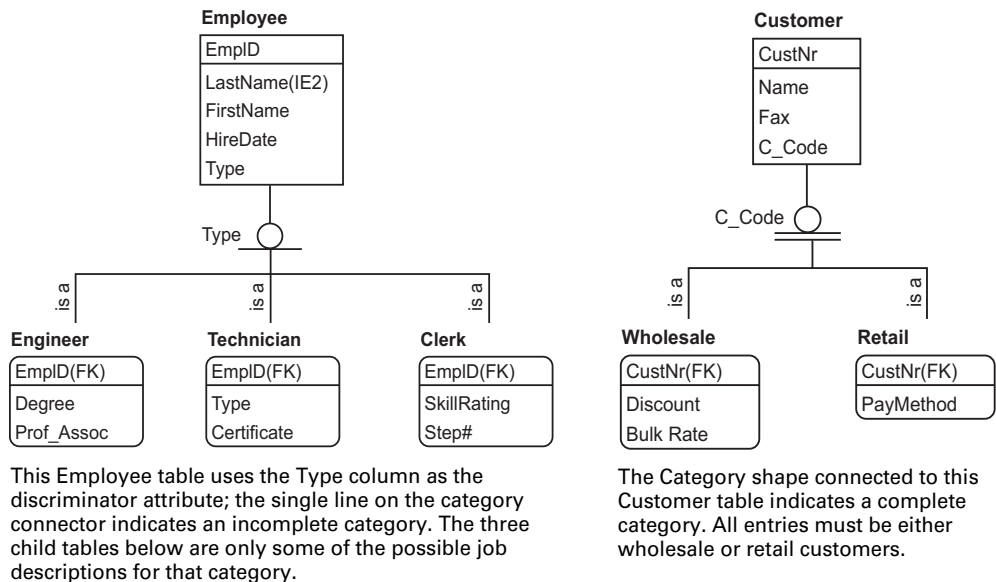
**TIP** You can set the Pushpin in the Table Properties dialog box if you want the dialog box to stay open while you edit a series of database components in the model.

## Using categories

Visio Enterprise supports **category tables**, which are subtypes of a parent table. For example, an employee's record (the generic parent) might contain a job-type column (the category discriminator); category entities in this example might include engineer, technician, and administrator. To define a category table, use the Category shape, the Parent To Category shape, and the Category To Child shape from the Entity Relationship or Object Relational stencil.

If a category table contains all possible subtypes, it is considered complete; if it contains only some possible subtypes, it is considered incomplete. Double lines in a Category shape indicate a complete category; a single line indicates an incomplete category. You can easily toggle a category between complete and incomplete, and you can use the Table Inheritance Properties dialog box to define the category's discriminator attribute on the parent table. Using a category establishes a foreign key relationship from the parent table to the child tables connected to the Category shape.

**Figure 8-3**  
Category tables





### **To define a category in a model:**

- 1 Drag a Category shape to your database model.
- 2 Choose the connector tool from the Standard toolbar, then click the Parent To Category shape. Position the connector tool over the center of the parent table so that the table is outlined, press the mouse button, then drag to the center of the Category shape. When the Category shape is outlined, release the mouse button.

Both connection points appear red.

- 3 With the connector tool still selected, click the Category To Child shape. Position the connector tool over the center of the Category shape so that it is outlined, press the mouse button, then drag to the center of the child table. When the child table is outlined, release the mouse button.

Both connection points appear red, and the primary keys from the parent table are added as foreign keys to the child table.

- 4 Repeat step 3 for each child table.

### **To set a Category connector shape as complete or incomplete:**

- 1 Right-click the Category connector shape you want to change, then choose Table Inheritance Properties from the shortcut menu.
- 2 In the Table Inheritance Properties dialog box, clear the Category Is Complete check box to mark the category as incomplete, or select the Category Is Complete check box to mark the category as complete.

An incomplete shape displays a single line below the circle.

A complete shape displays a double line below the circle.

### **To specify a category discriminator attribute:**

- 1 Right-click the Category shape you want to change, then choose Table Inheritance Properties from the shortcut menu.
- 2 On the Definition tab, click This Attribute, then choose a column from the parent table as the category discriminator.

# Working with columns

Visio Enterprise makes it easy to refine the tables in your database model by adding, editing, and deleting columns and their properties. You can add columns, remove columns, edit column properties, and change the order of the columns in a table.

**NOTE** For details about adding complex data types to database models that will be implemented on an object-relational DBMS, see [“Working with data types”](#) later in this chapter.

Columns in a table can be identified as a **primary key**; primary keys have unique values, which are used to identify instances of the table in a database, and each table must have a primary key in order to pass logical validation. Columns can also be identified as **foreign keys**. A foreign key describes the relationship of a child table to a parent table by defining the attribute from the parent table that the child table inherits.

Visio Enterprise automatically creates unique indexes on primary keys. Other columns can also be assigned a non-unique index called the **inversion entry**. Indexes optimize the performance of your database for data retrieval. You can edit the properties of an index in the Table Properties dialog box. For details about using indexes in a Visio Enterprise database model, see [“Creating and editing indexes”](#) later in this chapter.

**NOTE** You can edit default column name suffixes in the Database Modeling Preferences dialog box. Choose Database > Modeling Preferences, then click the Logical Misc tab. For details about setting modeling preferences, see “Customizing your database model diagram” in Chapter 7, “Creating database models.”

## To add a new column to a table:

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 In the database model diagram, right-click the table to which you want to add the column, then choose Edit Columns from the shortcut menu.
- 3 In the Columns In Table dialog box, click the Add button.  
A new column is added to the table using the default naming conventions you specified in the Modeling Preferences dialog box.
- 4 Click Close.



**To remove a column:**

- 1 Right-click the table from which you want to remove the column, then choose Edit Columns from the shortcut menu.
- 2 In the Columns In Table dialog box, select the column you want to remove, then click the Remove button.
- 3 Click Close.

**To reorder the columns in a table:**

- 1 Right-click the table in which you want to reorder the columns, then choose Edit Columns from the shortcut menu.
- 2 In the Columns In Table dialog box, select a column you want to reorder, then drag it up or down to move it to a new location in the column list.
- 3 Click Close.

**To edit or create a primary key:**

- 1 Right-click the table to which you want to add a primary key, then choose Table Properties from the shortcut menu.
- 2 In the Table Properties dialog box, click the Primary ID tab.
- 3 Select the column you want to be the primary key in the Available Columns list, and use the Move button (>>) to move it to the Primary ID list.  
*Optional* Customize the physical name and key type.
- 4 Click the Close button or press Alt+F4 to close the dialog box.

**NOTE** The position of primary keys in your table depends on the Order option you select on the Table tab in the Database Document Options dialog box (Database > Document Options). If you choose Physical Order, primary keys are displayed in the order they will have in the database. Otherwise, primary keys are displayed at the top of a table.

**To edit the properties of a column:**

- 1 Right-click the table with the column you want to edit, then choose Edit Columns from the shortcut menu.
- 2 In the Columns In Table dialog box, select the column you want to edit, then click the Edit button.
- 3 In the Column Properties dialog box, click the tab that contains the information you want to edit, and make your changes. The database model diagram updates automatically. You can choose Edit > Undo to reverse the changes.

Following is a brief description of each tab and the options it contains.

**Definition tab** Enter or edit a Physical Name to control the column's name in the database or a Conceptual Name to control the column's name as it appears in the database model. Check Allow NULL Values to make the column optional rather than mandatory.

**Data Type tab** Display either Portable or Physical Data Types, and assign Data Types to columns.

**Collection tab** Specify whether the column is a single value or contains a collection of information. Collections may be sets, lists, or multiple values, and may only be used in object-relational models.

**Check tab** Edit, add, remove, or preview check clauses which are included in the DDL script that you can create in the Generate Wizard or Update Database Wizard. For details, see [“Creating check clauses” in Chapter 9](#), [“Reverse engineering database code.”](#)

**Extended tab** Set DBMS-specific extended attributes for use with the Visio Enterprise database drivers.

**Notes tab** Type text to document the column. Text entered here appears in the Verbalizer window.



**NOTE** To get online help for an option on a tab, click the question mark button in the dialog box, then click the option for which you want help. Or right-click the option, then choose What's This.

- 4 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.



**TIP** You can set the Pushpin in the Column Properties dialog box if you want the dialog box to stay open while you edit a series of elements in the diagram.

## Working with data types

Visio Enterprise supports both relational and object-relational database models, so you can work with both **simple** and **composite data types**. Simple data types, used in both types of models, support one data value per column. Object-relational databases support composite data types, in which a column can contain multiple values or fields, and those fields can have different data types. The Visio Enterprise data type support ensures the database schemas you generate from your models are as efficient as possible.

When you use shapes from the Object Relational stencil to refine an object-relational database model, you can define composite data types and specify collection types for columns. Composite data types allow you to encapsulate fields that might contain multiple values of different data types. Collection types allow multiple instances of a given value to be stored in a field. Both composite data types and collection types can optimize the performance of a database by storing data in a single entity rather than using foreign keys and secondary tables.

## Setting data types

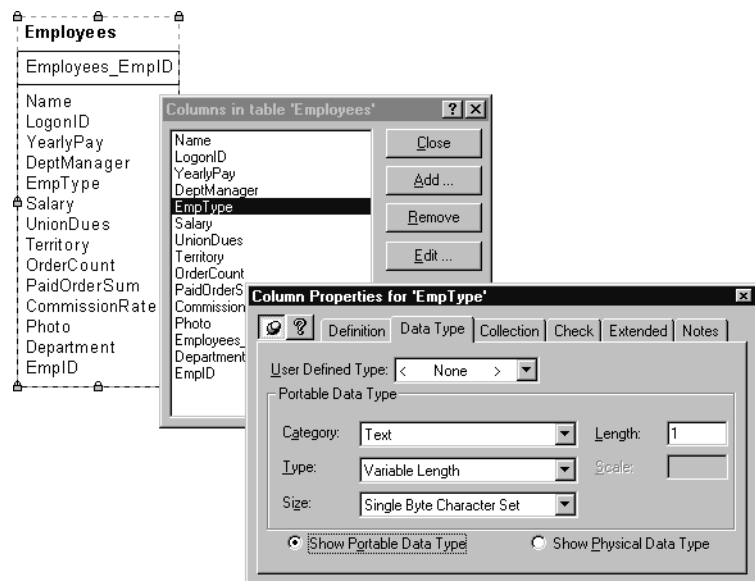
You can set data type properties for any column in the Column Properties dialog box. A column's data type specifies

- The kind of data that the column can contain.
- The types of operations you can perform on the data in the column.
- Column characteristics such as physical storage requirements.

With Visio Enterprise, you can use generic **portable data types**, which can be mapped to multiple databases at the time you generate a schema, or **physical data types**, which are specific to a target database. When you generate a database schema, Visio Enterprise maps portable data types to the closest data type supported by the DBMS you specify. This feature offers flexibility if you are designing a database model that will be implemented on more than one type of DBMS or if you are transferring a legacy database from one DBMS to another.

**Figure 8-4**

Each column in a table must be assigned a data type. In this example, the EmpType column is assigned a Variable Length Text portable data type.



When you work with portable data types, you can use the default specifications or define a user-defined data type for easy reuse. User-defined data types can save time and ensure consistency by providing a reusable data type with a standard definition. You choose Database > User-Defined Types to add, remove, and edit custom data types. You can also copy definitions from existing user-defined data types to new ones.

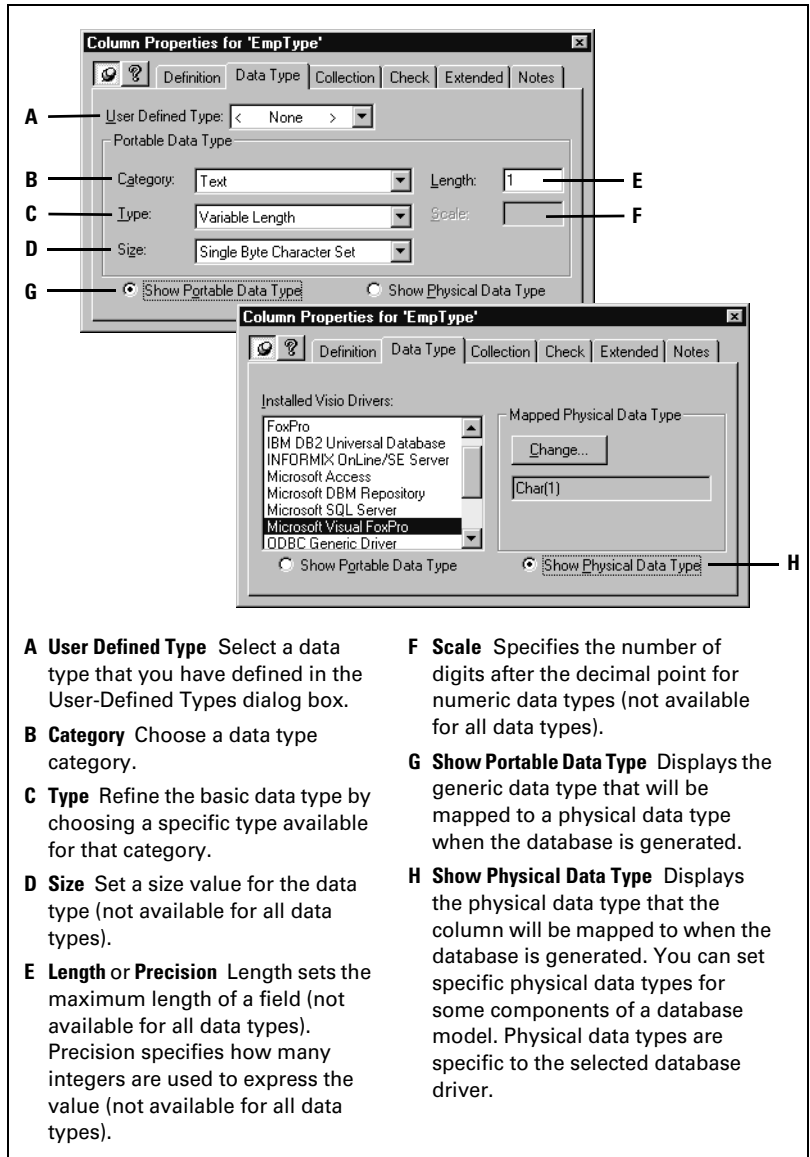
**NOTE** Before you set data type information in your database model, you should be familiar with the data types and storage requirements supported by your target DBMS. Refer to the documentation for your target DBMS to determine which data type to assign.

**To set the data type for a column:**

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Right-click the table containing the column whose data type you want to set, then choose Edit Columns from the shortcut menu.
- 3 Select the column you want to edit in the Columns In Table dialog box, then click the Edit button.
- 4 In the Column Properties dialog box, click the Data Type tab.
- 5 Select the data type options you want to set for the column.
- 6 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.

**Figure 8-5**

Data type options for a column



### To create a user-defined data type:

- 1 Choose Database > User Defined Types.

**NOTE** The Database command is available only when you start a model in Visio Enterprise using the Database Model template.

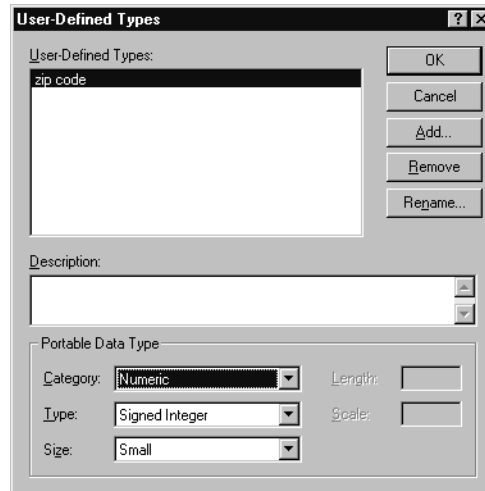
- 2 In the User-Defined Types dialog box, click Add.
- 3 In the Add New User Defined Type dialog box, enter a name.

*Optional* To base your new data type on an existing one, check Copy From, then select the name of an existing user-defined data type.

- 4 Click OK.
- 5 *Optional* Enter notes about the data type in the Description box.
- 6 Specify portable data type characteristics for the data type.
- 7 Click OK.

**Figure 8-6**

The characteristics you specify in the User Defined Types dialog box control how columns assigned that data type are mapped in the physical database.



### To remove a user-defined data type:

- 1 Choose Database > User Defined Types.

**NOTE** The Database command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 In the User-Defined Types dialog box, select the data type you want to delete.
- 3 Click Remove, then click OK.

**TIP** You can also delete a user-defined type from the Types window. To do so, choose Database > View > Type Selector, right-click the user-defined type, then choose Delete from the shortcut menu.

## Using collection types and composite data types in a database model

Collection types and composite data types can be used in models that will be implemented on object-relational databases. Collection types can be assigned to any column in any table, but unless you display physical data types in the model there is no way to see at a glance where they have been used. Composite data types appear as distinct elements in a diagram, and can be specified as the data type. You can use composite data types to create typed tables or views.

### To set a collection type:

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Right-click the table containing the column you want to specify as a collection type, then choose Edit Columns from the shortcut menu.
- 3 In the Columns In Table dialog box, select the column you want to set as a collection type, then click the Edit button.
- 4 In the Column Properties dialog box, click the Collection tab, then choose one of the following collection types:

**Unordered, Non-Repeating Group Of Values (Set)** Allows population instances of a field in a column to have multiple values, but each value must be unique. The values do not have any sorting criteria on them; therefore they are unordered.

**Ordered Group Of Values (List)** Creates an ordered collection of elements that allows duplicate elements. Each element in the collection has an ordinal position. An ordinal position is a position by which a value can be accessed. Because two values can be the same, they are differentiated by their ordinal positions.

**Unordered, Repeating Group Of Values (Multiset)** Creates a collection of elements that can have duplicate values. The elements do not have ordinal positions.

**Single Value (Column)** Specifies that no collection type property is applied. This is the default type.

- 5 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.

**To display collection types:**

- 1 Choose Database > Document Options.

**NOTE** The Database command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 In the Document Options dialog box, click the Table tab, then select Show Physical. The collection type name will precede the name of any column assigned a collection type.
- 3 Click OK.

**To create a composite data type:**

- 1 From the Object Relational stencil, drag a Type shape to your database model.
- 2 Right-click the Type shape, then choose Edit Fields from the shortcut menu.
- 3 In the Field In Type dialog box, click Add.
- 4 *Optional* To edit a field name, click the Edit button, click the Definition tab in the Field Properties dialog box, then enter a new name in the Physical Name field. To edit the other properties of the field, click the appropriate tab in the Field Properties dialog box and enter the information.
- 5 Click the Close button or press Alt+F4 to close the Field Properties dialog box, then click Close to close the Fields In Type dialog box.

**To add a composite data type to a table:**

- 1 Right-click a table in the database model and choose Edit Columns from the shortcut menu.
- 2 In the Columns in Table dialog box, click Add.
- 3 With the new column name selected, click Edit.
- 4 In the Column Properties dialog box click the Data Type tab, then choose the complex data type from the User Defined Type list.
- 5 *Optional* To edit the other properties of the column, click the appropriate tab in the Column Properties dialog box and enter the information.
- 6 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.



# Working with views

With a view, you can combine columns from several tables, and you can perform operations on the data from those columns. Views provide a convenient way to access information needed by the users of a database without changing the structure of the tables in the database. For example, a view for invoices might combine columns from a table containing customer contact information, another containing employee information, and a third containing order details. Because the data in the view is inherited from the columns in the other tables, database accuracy and efficiency are improved.

In Visio Enterprise, you can create views by dragging and dropping the View shape on the Entity Relationship or Object Relational stencil to your database model diagram. When you create a view using the View shape, Visio Enterprise automatically creates SQL code to define the view. You can also use the Code Editor to define DBMS-specific views. For details, see [“Working in the Code Editor” in Chapter 9, “Reverse engineering database code.”](#)

## To add a view to a model:

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 From either the Entity Relationship or Object Relational stencil, drag a View shape onto the model drawing page.

Visio Enterprise adds a view to the model. You can set view properties and add columns to define the view.

## To remove a view from the database model:

- 1 Select the view you want to remove, then press Delete.
- 2 In the Delete Object dialog box, choose Yes if you want to remove the view from the model and the Tables & Views window, or No to remove it from the drawing page but not from the model or the Tables & Views window.

**NOTE** If you changed the default modeling preferences for removing objects from the diagram to an option other than Ask User What To Do, the Delete Object dialog box will not appear. To check your settings for object removal, choose Database > Modeling Preferences, then click the Logical Diagram tab.

For details about setting Modeling Preferences, see [“Customizing your database model diagram” in Chapter 7, “Creating database models.”](#)

**To assign columns from tables to a view:**

- 1 Right-click the view to which you want to add columns, then choose Edit Columns from the shortcut menu.
- 2 In the Columns In View dialog box, click Add once for each column you want to add.
- 3 Select a column name and click Edit to define the column's properties.
- 4 In the View Column Properties dialog box, click the Source tab. Click Known Column In Another Table Or View, then click Change.
- 5 In the Pick A Column dialog box, choose a column from the list of tables and views in your model, then click OK.
- 6 Repeat steps 3 through 5 for each column you added.
- 7 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.

**To edit the properties of a column in a view:**

- 1 Right-click the view containing the columns you want to modify, then choose Edit Columns.
- 2 In the Columns In View dialog box, click to select the column you want to modify, then click Edit.
- 3 In the View Column Properties dialog box, click the tab that contains the information you want to edit, and make your changes. The database model diagram updates automatically. You can choose Edit > Undo to reverse the changes.

Following is a brief description of each tab and the options it contains.

**Source tab** Identify the source of the information for that column. Enter SQL code in the Definition Rule field to create a calculated value.

**Definition tab** Enter Physical and Conceptual Names for the view, and specify whether the column will accept NULL values.

**Data Type tab** Specify a data type for the column. Select Show Portable Data Type to assign a user-defined data type or a portable data type, or select Show Physical Data Type to assign a DBMS-specific data type.

**Notes tab** Type text to document the view. Text entered here appears in the Verbalizer window.

- 4 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.

### To edit the properties of a view:

- 1 Right-click the view whose properties you want to edit, then choose View Properties from the shortcut menu.

**TIP** You can also double-click the View shape to display the View Properties dialog box.

- 2 In the View Properties dialog box, click the tab that contains the information you want to edit, and make your changes. The database model diagram updates automatically. You can choose Edit > Undo to reverse the changes.

Following is a brief description of each tab and the options it contains.

**Definition tab** Enter Physical and Conceptual Names for the view, and specify a Name Space. To create a typed view, select a type from the Defining Type list.

**Join Criteria tab** Define the foreign key relationships for columns included in the view by joining columns from associated tables.

**SQL tab** View the SQL code that defines the view. Uncheck Auto-Generate to edit the code directly in the dialog box, or click Code Properties to open the Code Editor.

**Extended tab** Set DBMS-specific extended attributes for use with the Visio Enterprise database drivers.

**Notes tab** Type text to document the view. Text entered here appears in the Verbalizer window.

- 3 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.

### To edit a view's SQL code:

- 1 Right-click the view whose SQL code you want to edit, then choose View Properties from the shortcut menu.
- 2 In the View Properties dialog box, click the SQL tab.
- 3 Uncheck Auto-Generate. To edit the code, type in the SQL tab's code edit box, or click Code Properties to open the Code Editor.

On the Body tab of the Code Editor, you can edit the SQL code. On the Properties tab of the Code Editor, you can create a mirror file and choose a name for the view code, which appears in the Code window. Click OK when you are done.

For details about mirror files and using the Code Editor, see [Chapter 9, "Reverse engineering database code."](#)

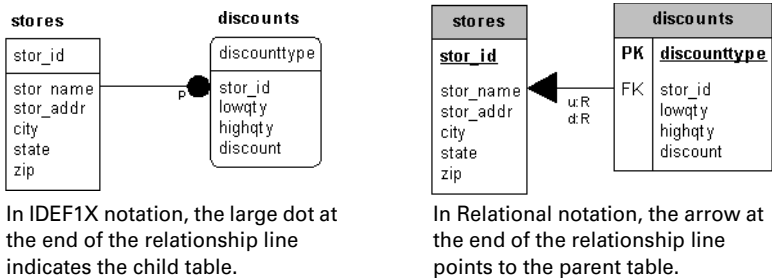
- 4 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In View dialog box.

# Working with relationships

As you refine your database model, you can add and remove relationships between tables, and edit relationship properties. Relationship shapes are one-dimensional connectors that you can glue between two table shapes to define a relationship between them. Relationships indicate visually how the tables in your database model interact with one another, and editing their properties enables you to control how data is stored and retrieved in the database. Relationships in a model are characterized in terms of direction, referential integrity, and cardinality.

The originating table in a relationship is known as the parent; the receiving table that assumes or inherits attributes from the parent table is known as the child. When you add a relationship to your model, you connect first to the parent table and then to the child table in order to establish the relationship's direction. Attributes inherited from the parent table appear as foreign keys in the child table. Setting a parent/child relationship between tables enables you to specify referential integrity, which establishes the rules that govern what effect a change to the parent table has on the child table. The cardinality of a relationship defines the ratio of instances of a data value in the child table that can exist in proportion to a related data value in the parent table.

**Figure 8-7**  
The directionality of relationships is displayed differently depending on a model's notation.



**NOTE** The notation for your model determines how the relationship's directional is displayed. In Relational notation, an arrow points to the parent table. In IDEF1X notation, and when Drawn With Crow's Feet is selected on the Relationship tab of the Database Document Options dialog box, a dot or crow's foot symbol indicates the child table.

## To add a relationship:

- 1 Choose File > New > Database > Database Model to start a model in Visio Enterprise using the Database Model template. Add at least two table entities to your model.
- 2 Choose the connector tool from the toolbar. Position the connector tool over the center of the parent table so that the table has a heavy outline surrounding it, press the mouse button, and drag the shape to the center of the

child table. When the child table also displays a heavy outline, release the mouse button. Both connection points should appear red, and any primary keys in the parent table are added as foreign keys to the child table.

**TIP** Alternatively, you can drag a Relationship shape onto the drawing page. Hold the Ctrl key while you drag one end of the Relationship shape to the Entity shape, then release the mouse. Repeat for the other end of the Relationship shape.

For details about defining the relationship, see [“To edit the properties of a relationship:”](#) below.

### To remove a relationship:

- Click to select the relationship you want to delete, then press the Delete key.

### To edit the properties of a relationship:



- 1 Choose the pointer tool from the Standard toolbar.
- 2 Right-click the relationship whose properties you want to edit, then choose Relationship Properties from the shortcut menu.

**TIP** You can also double-click the relationship shape to display the Relationship Properties dialog box.

- 3 In the Relationship Properties dialog box, click the tab that contains the information you want to edit, and make your changes. The database model diagram updates automatically. You can choose Edit > Undo to reverse the changes.

Following is a brief description of each tab and the options it contains.

**Definition tab** Associate columns in the parent and child tables and enter role names for foreign keys.

**Name tab** Enter a verb phrase, inverse phrase, physical name, or notes about the relationship.

**Miscellaneous tab** Choose the type of cardinality, whether the relationship is identifying or nonidentifying, and whether the relationship is optional.

**Ref. Integrity tab** Choose options that determine the effect on the child table when information in the parent table is updated or deleted.



**NOTE** To get online help for an option on a tab, click the question mark button in the dialog box, then click the option for which you want help. Or right-click the option, then choose What’s This.

- 4 Click the Close button or press Alt+F4 to close the dialog box.

For details about how to edit or define referential integrity and extended attributes, see [“Creating and editing indexes”](#) and [“Setting extended attributes”](#) later in this chapter.

## Setting referential integrity and cardinality

The principle of referential integrity requires that the foreign key values of a child table are either NULL or match exactly those of the primary key in the parent table in order to ensure validity in the physical implementation of the database. For example, a referential integrity rule may require that all employees be assigned to one of the company departments that are stored in another table. This way, no employee can be assigned to a department that does not exist.

To maintain referential integrity, you can specify rules that govern how a change to the data values in one table in the relationship affects the data values in the other table. When you enforce referential integrity, the foreign key (FK) relationships between related tables remain valid when you insert, update, or delete records.

You can edit referential integrity settings in the Relationship Properties dialog box; Visio Enterprise automatically generates referential integrity constraints between related tables using the settings you define. You can also choose to display referential integrity notation next to the relationship lines in your model. Visio Enterprise displays the notation in this form: u:C, where “u” is the type of change (update or delete) and “C” is the type of action (No Action, Cascade, or Set NULL).

<i>Referential integrity notation</i>	<i>Triggered action</i>
u	On update
d	On delete
R	No action
C	Cascade
N	Set NULL
(Blank)	Do not enforce referential integrity

Visio Enterprise also supports **cardinality** notation, which describes how many instances of a value in a child table connect to how many instances of a value in a parent table.

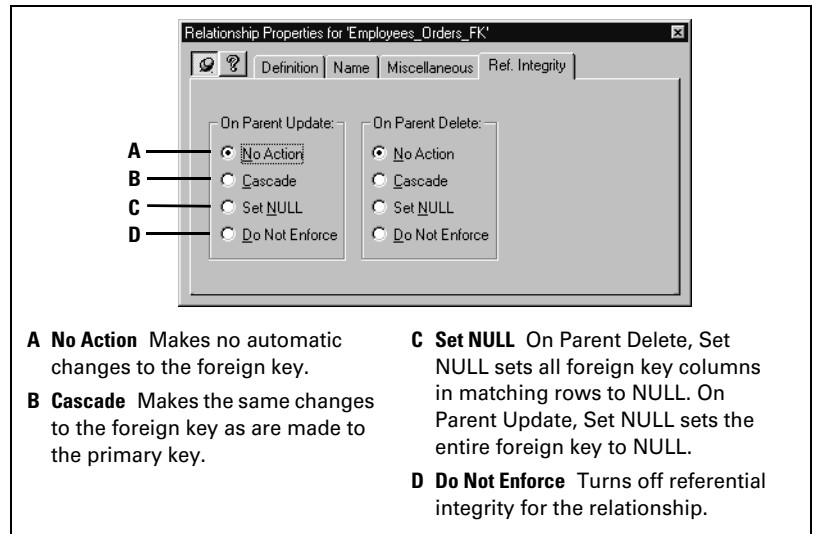
**NOTE** Visio Enterprise does not support many-to-many relationships in IDEF1X notation.

### To edit referential integrity settings:

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Right-click the Relationship shape whose referential integrity settings you want to edit, then choose Relationship Properties from the shortcut menu.  
**TIP** You can also double-click the relationship to display the Relationship Properties dialog box.
- 3 In the Relationship Properties dialog box, click the Ref. Integrity tab.
- 4 Select On Parent Update and On Parent Delete options.
- 5 Click the Close button or press Alt+F4 to close the dialog box.

**Figure 8-8**

When a record in a database is updated or deleted, you can choose any of the actions shown.



**NOTE** Not all DBMSs support all the options available on the Ref. Integrity tab of the Relationship Properties dialog box. Refer to the documentation for your target DBMS when setting referential integrity options. You should choose the settings that your DBMS supports.

### To show referential integrity settings in a database model:

- 1 Choose Database > Document Options.

**NOTE** The Database command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the Relationship tab, check Referential Integrity, then click OK.

For details about setting referential integrity in Visio Enterprise, see the next section, [“Creating and editing indexes.”](#)

### To show cardinality notation in your database model:

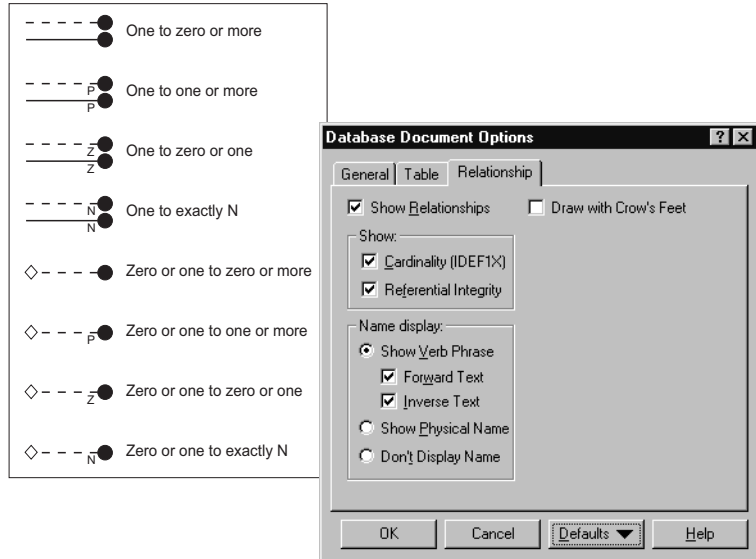
- 1 Choose Database > Document Options.

**NOTE** The Database command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the Relationship tab, check Cardinality (IDEF1X), then click OK.

**Figure 8-9**

When a model uses IDEF1X notation, you can check Cardinality (IDEF1X) on the Relationship tab of the Database Document Options dialog box to display cardinality settings. To the right are examples of cardinality notation.



**NOTE** Visio Enterprise Relational notation supports cardinality, but does not display it graphically.

## Creating and editing indexes

An index helps your database system find and sort records faster. Creating indexes for columns you plan to search frequently can significantly improve your database's performance. You can define indexes manually, and you can modify the indexes that Visio Enterprise creates automatically. For example, **primary keys** (PK) automatically generate a unique constraint. In an IDEF1X model, **alternate keys** (AK) generate a unique index, and **inversion entries** (IE) generate a non-unique index or a constraint. When Visio Enterprise generates an index for you it automatically assigns it a name.



**NOTE** Some database applications have a limit on the type and number of indexes you can create per table. For specific information, see your DBMS documentation. Visio Enterprise checks for this limit during physical validation. For details about validating your database model, see [“Validating a database model”](#) later in this chapter.

You can define indexes and edit their properties using the Table Properties dialog box. For example, you can edit the name of an index, change index options, rearrange the order of columns in an index, or set extended attributes for an index. Extended attributes for an index are driver-specific and are not supported by all DBMSs.

**To create an index:**

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Right-click the table to which you would like to add an index, then choose Table Properties from the shortcut menu.

**TIP** You can also double-click the table shape to display the Table Properties dialog box.

- 3 In the Table Properties dialog box, click the Indexes tab, then click New.
- 4 In the Create Index dialog box, enter a name for the index, then click OK.
- 5 From the Available Columns list, select the name of each column that you want to include in this index, then click the Move (>>) button.
- 6 Check the Asc check box to create an index with ascending sort order, or clear it to create one with descending sort order.
- 7 From the Index Type drop-down list, choose an option to create a unique or non-unique index.
- 8 *Optional* Click the Options button to set driver-specific extended index attributes.
- 9 Click the Close button or press Alt+F4 to close the dialog box.

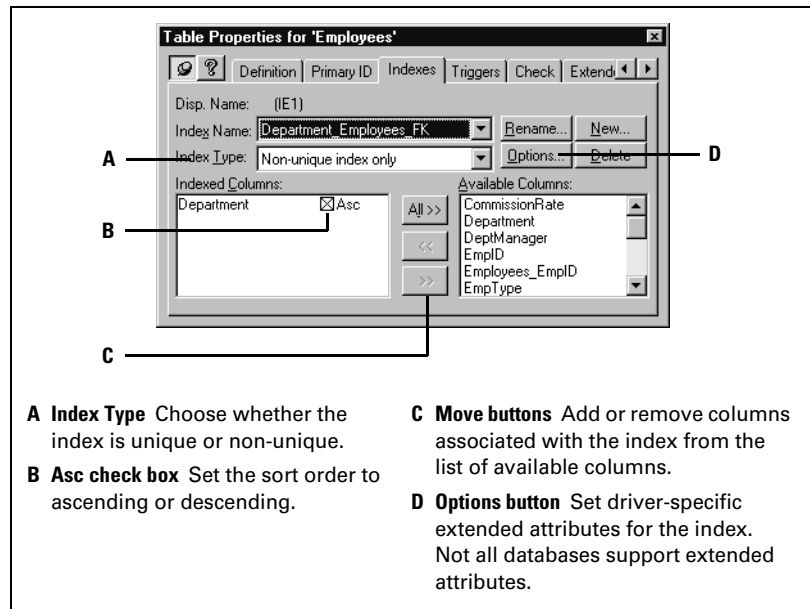
**To remove an index:**

- 1 Right-click a table containing indexed columns, then choose Table Properties from the shortcut menu.
- 2 In the Table Properties dialog box, click the Indexes tab.
- 3 In the Index Name list, select the name of the index you want to delete, then click Delete.
- 4 Click the Close button or press Alt+F4 to close the dialog box.

### To edit index properties:

- 1 Right-click a table containing indexed columns, then choose Table Properties from the shortcut menu.
- 2 In the Table Properties dialog box, click the Indexes tab.
- 3 In the Index Name list, select the name of an index.
- 4 Edit the properties of the index. For example, to enter a new name for the index, click Rename. To choose a unique or non-unique index, choose an option in the Index Type box.
- 5 Click the Close button or press Alt+F4 to close the dialog box.

**Figure 8-10**  
Options for defining an index



### To add, remove, or reorder columns in an index:

- 1 Right-click a table containing indexed columns, then choose Table Properties from the shortcut menu.
- 2 In the Table Properties dialog box, click the Indexes tab.
- 3 In the Index Name list, select the name of the index you want to edit.
- 4 Do one of the following:
  - To add a column to an index, select the name of the column in the list of Available Columns, then click the Move (<<) button.
  - To remove a column from an index, select the name of the column in the list of Indexed Columns, then click the Move (>>) button.

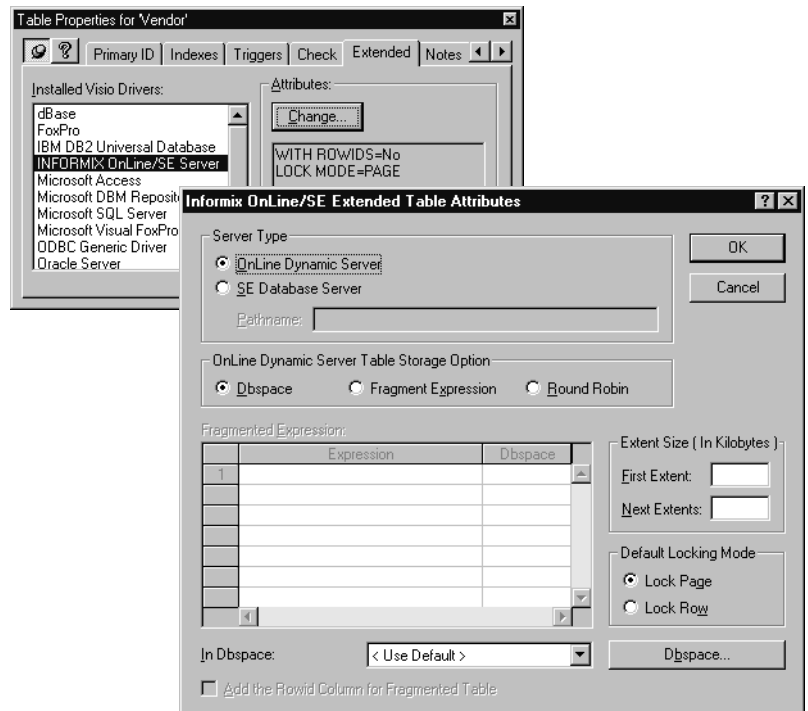
- To reorder a column, select the name of the column in the list of Indexed Columns, hold down the mouse button, then drag the column name to its new position in the list.
- 5 Click the Close button or press Alt+F4 to close the dialog box.

## Setting extended attributes

You can set **extended attributes** for a table, column, or relationship using the Table Properties, Column Properties, or Relationship Properties dialog box. Extended attributes are specific to a target DBMS, and allow you to control more precisely the way elements defined in your model are mapped to a target database.

**Figure 8-11**

Using the Extended tab, you can specify DBMS-specific attributes for a table. The attributes you can set depend on the capabilities supported by the DBMS.



For example, with Informix you can use extended attributes to choose dynamic storage options for tables. Similarly, if you are targeting your model to SQL Server 6.5, you can specify the segment of the target database on which a table will be stored.

**To set extended attributes for a table or view:**

- 1 Open the Visio Enterprise Database Model template file that represents your database model.
- 2 Right-click the table or view whose extended attributes you want to define, then choose Table Properties or View Properties from the shortcut menu.

**NOTE** Microsoft Repository drivers support extended attributes for views.

- 3 In the Table Properties or View Properties dialog box, click the Extended tab.
- 4 From the Installed Visio Drivers list, select a Visio Enterprise database driver that matches the DBMS you plan to use.

For example, if you want to generate a database schema for a Sybase database, choose the Visio Enterprise Sybase driver.

- 5 Click the Change button.

Visio Enterprise opens a driver-specific extended attributes dialog box.

**NOTE** If a particular DBMS does not support extended attributes, Visio Enterprise dims the Change button on the Extended tab, and the option is unavailable.

- 6 Fill in the appropriate information and choose any options that you want.
- 7 Click OK.
- 8 Click the Close button or press Alt+F4 to close the dialog box.

**To set extended attributes for a column:**

- 1 Right-click the table containing the column whose extended attributes you want to define, then choose Edit Columns from the shortcut menu.
- 2 In the Columns In Table dialog box, select a column to edit, then click Edit.
- 3 In the Column Properties dialog box, click the Extended tab.
- 4 From the Installed Visio Drivers list, select a Visio Enterprise database driver that matches the database you plan to use.

For example, if you want to generate a database schema for a Sybase database, choose the Visio Enterprise Sybase driver.

- 5 Click the Change button.

Visio Enterprise opens a driver-specific extended attributes dialog box.

**NOTE** If a particular DBMS does not support extended attributes, Visio Enterprise dims the Change button on the Extended tab, and the option is unavailable.

- 6 In the dialog box that appears, specify the extended attribute information for the DBMS, then click OK.
- 7 Click the Close button or press Alt+F4 to close the Column Properties dialog box, then click Close to close the Columns In Table dialog box.

# Validating a database model

You can choose to check the validity of your model at any time during the modeling process. Visio Enterprise checks your database model for both logical and physical validity and displays progress notes, warnings, and errors in the Output window. Using the information in the Output window you can quickly navigate to any warning conditions or errors in the model. When you use the Generate, Update Model, and Update Database Wizards, Visio Enterprise automatically validates your database model before taking any action. For details about using these wizards, see [Chapter 11, “Generating a database schema,”](#) and [Chapter 12, “Updating models and databases.”](#)

During logical validation Visio Enterprise checks your model for adherence to generally accepted database design standards. For example, circular table relationships produce an error during logical validation; similarly, using the same physical name for two columns in a table produces an error. During physical validation, Visio Enterprise checks your database model for compliance with DBMS-specific requirements. For example, an Oracle DBMS allows only one column in a table to use the Long data type; if a table contains more than one column using this data type, validation will generate an error. It is possible to have a logically correct database model that contains physical errors, especially if you are reverse engineering a legacy database to a new DBMS.

When you use the Database > Model > Check command to validate a model, Visio Enterprise displays any progress notes, errors, or warnings in the Output window. If the model contains any critical errors, you must fix them before you can use the Generate, Update Model, Update Database, or Transfer commands. Warnings do not prevent you from using these commands.

## To validate a database model:

- Choose Database > Model > Check.

**NOTE** The Database command is available only when you start a model in Visio Enterprise using the Database Model template.

**To correct an error or warning:**

- 1 In the Output window, double-click the error or warning text.  
Visio Enterprise selects the structure in the diagram containing the error.
- 2 Right-click the error text in the Output window, then choose Get Help For Message from the shortcut menu. Or select the text, then press F1.  
Visio Enterprise displays an explanation for the error and how to resolve it.
- 3 Make the necessary correction.
- 4 Repeat steps 1–3 for each error or warning in the Output window.

## Next steps

After you have refined and validated the logical model, you are ready to generate or update a database schema. You may also want to create a report, so you can review all tables or types in a design or see a summary of the model.

<i>If you want to</i>	<i>See</i>
Generate a new database schema	<a href="#">Chapter 11, “Generating a database schema”</a>
Update the schema of an existing database	<a href="#">“Running the Update Database Wizard” in Chapter 12, “Updating models and databases”</a>
Synchronize a model with the database from which it was extracted	<a href="#">“Running the Update Model Wizard” in Chapter 12, “Updating models and databases”</a>
Create a report	<a href="#">Chapter 13, “Creating reports”</a>

# Reverse engineering database code

A database model can include code in the form of stored procedures, triggers, check clauses, and so on. This code is preserved when you reverse engineer a database, so that you can work with code components of your model in Visio® Enterprise. As you refine your database model, you can also create new code, and view and edit the code that was extracted by the Reverse Engineer Wizard. If you are reverse engineering a database to a new platform, you can transfer the code as well and view platform-specific implementations of code as you work. In addition, Visio Enterprise supports the use of version control software, so that you can track revisions to all the code associated with your database model.

This chapter describes the tools you use to reverse engineer code in a database model. In this context, “code” refers to stored procedures, functions, view definitions, triggers, check clauses, and other platform-specific code that you can view and edit in the Database Model solution.

## Topics in this chapter

- Viewing and editing code ..... 176
- Working in the Code Editor..... 178
- Adding code to a model..... 180
- Saving and synchronizing code ..... 186
- Transferring code to a new target platform ..... 188
- Next steps..... 189

# Viewing and editing code

A database model can include code associated with objects in the database or the database itself. When you reverse engineer a database that contains code, such as stored procedures, Visio Enterprise lists each code component in the order it was extracted in the Code window. Later, if you run the Generate Wizard or Update Database Wizard to create a DDL (Data Definition Language) script of the changes you've made to the model, the code is generated in the script in the same order as it appears in the Code window.

As you work in a model, you can create, view, and edit **platform-specific** or **portable** code associated with a database or its components. Only platform-specific code appears in the Code window, which includes all the code extracted from a database and any new code you write that is intended for a particular database, which you specify by setting the current database driver for the model to that database. Portable code is generic SQL generated by Visio Enterprise and not associated with any database driver. After you generate or update a database for a model that includes portable code, Visio Enterprise generates a platform-specific copy of the code for that database, which you can view in the Code window.

The Code window designates each of the following types of platform-specific code with a unique icon:



**Stored procedures** User-defined functions and other procedures that can be invoked with a CALL statement and are stored in an executable format in the database.



**Function** Procedural code that returns a value.



**Triggers** Code associated with specific database tables that is invoked automatically whenever rows of the associated table are inserted, updated, or deleted.



**Check clauses** Code applied to a table or column that enforces constraints.



**View Definitions** Code usually containing SELECT statements that assigns the results of a query to a named, table-like structure called a view.



**Raw DDL** Other code that does not fall into one of the above categories; for example, synonyms or package constructs in an Oracle database.



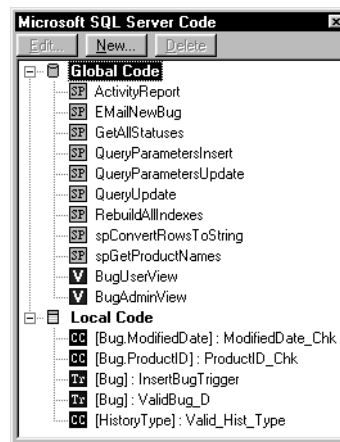
Not all platforms support these types of code. For example, although you can define a new view in your database model, you cannot use the model to generate a Corel Paradox database with a view, because Paradox does not support views. You need to know the capabilities of your target database when creating code for a model.

A model can be used to generate different target databases, but the Code window always displays a platform-specific version of the code based on the currently selected database driver. You can have multiple versions of the same code in your model to support the different databases you intend to generate, but the Code window shows only one at a time. To associate code with a different target database, you **transfer** it to the intended platform. You can switch between code implementations displayed in the Code window by changing the current driver. For details, see “[Transferring code to a new target platform](#)” later in this chapter.

Code is shown in the Code window as **global** or **local**. Global code refers to any stored procedure, function, view definition, or raw DDL code that is not associated with a specific table. Local code refers to triggers or check clauses for a specific table or column in your model.

**Figure 9-1**

The Code window lists all the stored procedures, triggers, functions, and other code associated with a particular database driver for your model.



The name of the current default driver appears in the title bar.

### To open the Code window:

- Choose Database > View > Code Selector.

#### Options in the Code window

<i>If you want to</i>	<i>Do this in the Code window</i>
Reorder code items in the list	Drag an item to a new position in the list.
Remove code from your model	Select an item in the list, then press Delete.
Display only global code	Right-click any item, then choose Show Only Global Code.
Print the code for an item	Right-click an item in the list, then click the Print button.
Create a new stored procedure, function, view definition, or other DDL code	Click New, then work in the Code Editor.
View an item in the Code Editor	Select an item in the list, then click Edit. Or double-click the item.



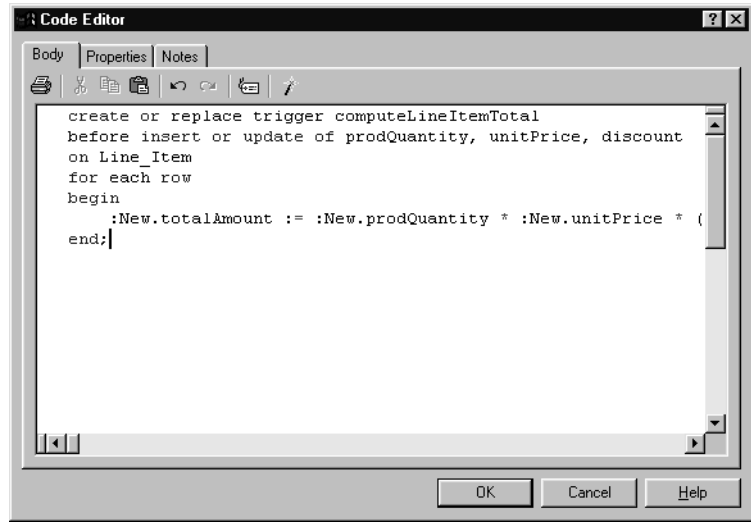
## Working in the Code Editor

To view or edit the code shown in the Code window, you can display the Code Editor, which has a toolbar for common editing tasks and options for changing the window's appearance.

You can customize the editing environment to suit your working style. The Code Editor supports drag-and-drop text editing as well as an unlimited number of undo and redo instances. Toolbar buttons provide easy access to the Cut, Paste, Undo, and Redo commands. Pause the pointer over a button to see what it does. In addition, you can change the colors used in the Code Editor, specify indentation preferences, and more by changing the window properties.

**Figure 9-2**

As you work in the Code Editor, you can set options to automatically indent code, color keywords, and insert blank code skeletons.



**To open code in the Code Editor:**

- 1 If the Code window is not already open, open it by choosing Database > View > Code Selector.

**NOTE** This command is available only when you start a model with the Database Model template.

- 2 Click a code item in the list, then click Edit. Or double-click a code item in the list.
- 3 View and revise code on the following tabs:

**Body tab** Enter and edit SQL statements. To edit, use the toolbar buttons, or right-click in the code to see additional options.

**Properties tab** Enter or revise the code component's name, choose the type of code it represents, and create a mirror file for saving the code component externally.

**Notes tab** Add code comments that will be generated in the DDL script for the model.



- 4 To set preferences for working in the Code Editor, click the Window Properties button. Or right-click in the code, then choose Properties from the shortcut menu.

- 5 In the Window Properties dialog box, choose options from the various tabs. The table below shows commonly used options.
- 6 When you are finished, click OK to return to the Code Editor. To close the Code Editor, click OK.

#### Working in the Code Editor

<i>If you want to</i>	<i>Do this</i>
Automatically indent lines as you enter code to follow the scoping rules of SQL	Right-click in the code, then choose Properties. On the Language/Tabs tab, choose Follow Language Scoping, then click OK.
Assign keyboard shortcuts	Right-click in the code, then choose Properties. On the Keyboard tab, choose a command, then a new key assignment.
Change spaces to tab characters	Select the code, right-click the selection, then choose Tabify.
Change keyword colors in the code	Right-click in the code, then choose Properties. On the Color/Fonts tab, choose an item, then choose a new foreground or background color.

## Adding code to a model

You can define new stored procedures, triggers, views, and other code as you work in your database model. To add global code to your model quickly, you use the Code window to create a new stored procedure, function, view definition, or other platform-specific DDL code. To create new local code, you work with the Properties dialog boxes for the different shapes.

When you create new code for a model, the code is platform-specific. Visio Enterprise associates the code with the database driver currently selected in the Database Drivers dialog box (Database > Drivers). You can later migrate the code, or a copy of it, to a different platform. For details, see [“Transferring code to a new target platform”](#) later in this chapter.

After you write the code and define its properties in the Code Editor, it appears as an item in the Code window. If you later generate a DDL script from your model, the new code appears in the script in the order it appears in the Code window. For details about DDL scripts, see [“Generating and editing a DDL script”](#) in Chapter 11, [“Generating a database schema,”](#) or [“Creating a script that updates a database”](#) in Chapter 12, [“Updating models and databases.”](#)

## Creating stored procedures, functions, and other global code

Stored procedures, functions, and other platform-specific types of DDL code are referred to as global code in Visio Enterprise. You can work in the Code Editor to write new code of any of these types by entering SQL statements and defining properties for the code item. Visio Enterprise adds new global code to the list of code items in the Code window.

Your target database may support types of code other than stored procedures and functions. For example, if your target platform is an Oracle database, you can create a package to group together related variables, exceptions, and so on. In your database model, you can create such platform-specific code types by specifying Raw DDL as the code type. When you use the Generate Wizard or Update Database Wizard to generate a DDL script for a model that contains code marked as Raw DDL, the wizard outputs whatever you defined in the Body tab of the Code Editor.

### To create a stored procedure, function, or other global DDL code:

- 1 If the Code window is not already open, choose Database > View > Code Selector.

**NOTE** This command is available only when you start a diagram with the Database Model template.

- 2 Click New to open the Code Editor.
- 3 On the Properties tab, type a name for the code item. Under Type, choose Stored Procedure, Function, or Raw DDL.
- 4 On the Body tab, enter SQL statements. When you're finished, click OK.  
The code is added to the Global Code list in the Code window.

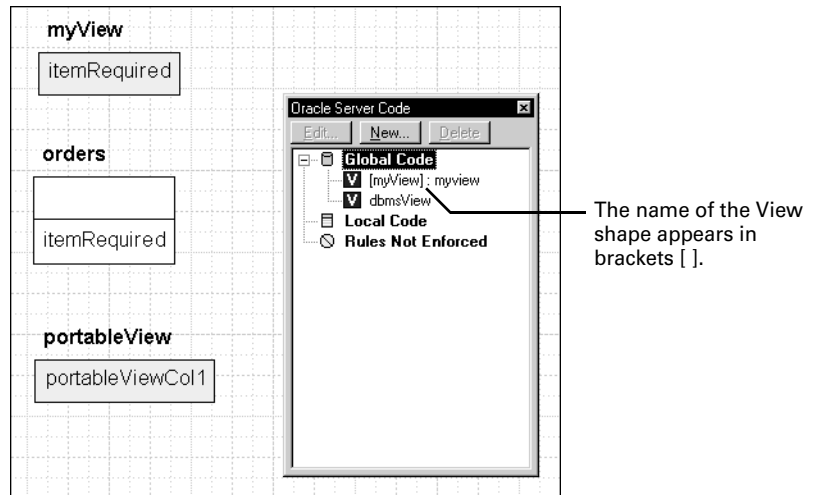
## Defining view code

In Visio Enterprise, you can work directly in the Code Editor to define a view intended for a particular target platform. When you define a view in this manner, the view is treated like any other code. The view is associated with the current database driver and is not visually represented in your model except as an item in the Code window.

When you reverse engineer a database that contains views, they appear in the Tables & Views window, so that you can drop a view into your model, where it is represented by the View shape. Because a view is defined by SQL code, it also appears in the Code window.

**Figure 9-3**

Views can represent platform-specific or portable code. Only platform-specific views appear in the Code window. Here, the PORTABLE view was created by using the View shape, then using the Auto-Generated option, which generates SQL code automatically for the view.



For most target databases, you create a new view in the Code Editor by using the CREATE VIEW statement to name the view that contains the records specified by the SELECT statement. For example, the following code creates the view MYVIEW that displays the ITEMREQUIRED record from the ORDERS table:

```
CREATE VIEW myview AS
SELECT itemRequired FROM ORDERS;
```

If you want to create a view that also appears in your model on the drawing page, you can use the View shape. With the View shape, you can quickly define a simple view for which Visio Enterprise can automatically generate SQL code that defines the view's contents. Or you can overwrite the generated code for a View shape by disabling the Auto-Generated option in the SQL tab of the View Properties dialog box, and then typing in the Code Editor. When you customize the code for a View shape, Visio Enterprise associates the view definition with the currently selected database driver, and the view code appears in the Code window for that driver. For details about working with the View shape, see [“Working with views” in Chapter 8, “Refining database models.”](#)

#### **To define a view:**

- 1 If the Code window is not already open, choose Database > View > Code Selector.

**NOTE** This command is available only when you start a diagram with the Database Model template.

- 2 Click New to open the Code Editor.
- 3 On the Properties tab, type a name for the code item. Under Type, choose View Definition.
- 4 On the Body tab, enter SQL statements. When you're finished, click OK.  
The code is added to the Global Code list in the Code window.

If you create a view based on another view, you must make sure that the original view appears in the Code window before any views based on it. This is because Visio Enterprise generates the code in a DDL script in the order that code items appear in the Code window.

#### **To reorder views in the Code window:**

- 1 If the Code window is not already open, choose Database > View > Code Selector.

**NOTE** This command is available only when you start a diagram with the Database Model template.

- 2 Drag the view you want to reorder to a new position in the window.

## Creating trigger code

Because a trigger is associated with a specific database table, you create a trigger in Visio Enterprise by working in the Table Properties dialog box. When you reverse engineer a database that contains triggers, or add a new trigger to a table in your model, the trigger code is shown on the Triggers tab of the Table Properties dialog box and is listed under Local Code in the Code window.

A trigger can be executed—or triggered—when specific database events occur, such as when a record is inserted, updated, or deleted. For example, you can use a trigger to log audit information to a separate table, calculate derived information such as the extended cost of price times quantity, notify logged-on users of limits that have been exceeded, or call a stored procedure.

### To create a trigger for a table:

- 1 Right-click a table, then choose Table Properties from the shortcut menu.
- 2 On the Triggers tab, click Add to open the Code Editor.
- 3 On the Properties tab, type a name for the trigger.
- 4 On the Body tab, enter SQL statements. When you're finished, click OK to return to the Table Properties dialog box.

The trigger is added to the Local Code list in the Code window. You can now close the Table Properties dialog box, or refine other table properties.

## Creating check clauses

You can use a check clause to enforce constraints on a table or column. The Code window lists all check clauses extracted from a database you reverse engineer as local code. You can also create a new check clause by working in the Table Properties or Column Properties dialog box.

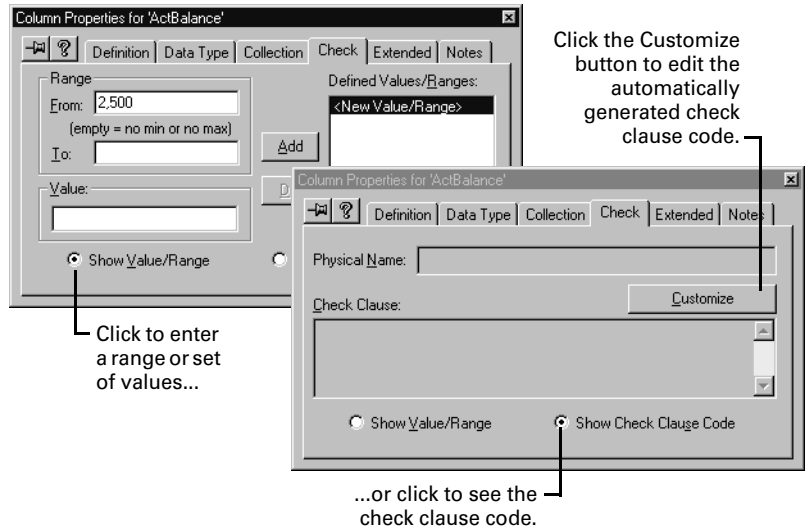
When you create a new check clause for a table, the code is specific to the platform currently selected in the Database Drivers dialog box and appears in the Code window for that driver. When you create a new check clause for a column, you can define the check clause yourself for your intended target database, or have Visio Enterprise automatically generate one based on a range or set of values that you specify. Automatically generated check clause code for a column does not appear in the Code window. If you then generate or update a DDL script for your target database, Visio Enterprise adds the check clause to the Code window for that database.



For example, you can add a check clause to the ActBalance record in the Customers table to check a customer's account balance, as Figure 9-4 shows. You can specify a range of values to be checked, and then Visio Enterprise will automatically generate a check clause. To customize the check clause for a particular database, you can edit the code. Visio Enterprise then adds the check clause to the Code window for the currently selected database driver.

**Figure 9-4**

Visio Enterprise can automatically generate a check clause for a range or set of values you specify. Or you can customize a check clause for the intended target database.



#### To create a check clause for a table:

- 1 Right-click a table, then choose Table Properties from the shortcut menu.
- 2 On the Check tab, click Add to open the Code Editor.
- 3 On the Properties tab, type a name for the check clause.
- 4 On the Body tab, enter SQL statements. When you're finished, click OK to return to the Table Properties dialog box.

The check clause is added to the Local Code list in the Code window. You can now close the Table Properties dialog box, or refine other table properties.

**To create a check clause for a column:**

- 1 Right-click a table, then choose Edit Columns from the shortcut menu.
- 2 Select a column from the Columns In Table list, click Edit, then click the Check tab.
- 3 To generate a check clause based on a range of values automatically, click Show Value/Range, then enter values in the From and To boxes to define the range of the constraint. Click Add.
- 4 To generate a check clause for a set of values automatically, type a value in the Value box, then click Add. Repeat for additional values, clicking Add after each.
- 5 To customize or view the check clause code, click Show Check Clause. To modify the check clause, click Customize, then type in the Check Clause box. The customized check clause is added to the Local Code list in the Code window.
- 6 To close the Column Properties dialog box, click the Close box. Or keep it open and refine other column properties.
- 7 Click Close to close the Columns in Table dialog box.

**NOTE** An automatically generated check clause does not appear in the Code window, but a customized check clause does.

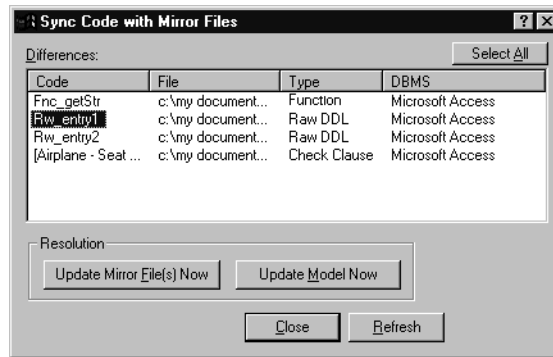
## Saving and synchronizing code

You can save the code you add to a database model diagram as a file separate from your model. By creating a **mirror file**, you create a file of your stored procedure, trigger, or check clause that you can save, work with outside of Visio Enterprise, and store using your version control software.

Once you've created a mirror file of your code, you can keep it in sync with your model by choosing Database > Code > Synchronize. Then if you edit the code outside of Visio Enterprise, or change it while working in the Code Editor, the changes are synchronized between the model and the mirror file.

**Figure 9-5**

You can create a mirror file for any code in your model, so that it can be stored in your version control software. To synchronize the model code with its mirror file, choose Database > Code > Synchronize.



**To save code in a mirror file:**

- 1 Choose Database > View > Code Selector to open the Code window, if it is not already open.
- 2 In the Code window, select the code you want to save, then click Edit. Or double-click the code.  
The Code Editor opens.
- 3 Click the Properties tab. In the File box, enter a path and file name, or click Browse to locate the file you want, then click OK.
- 4 In the Code Editor, click OK.

**To synchronize code in a model with a mirror file:**

- 1 Choose Database > Code > Synchronize.

The Synchronize dialog box opens. If a change has been made to any stored procedure, trigger, or check clause in the model, the code is listed under Differences.

- 2 Select the stored procedure, trigger, or check clause you want to synchronize, then choose one of the following:

**Update Mirror File(s) Now** Click to change the code stored in the mirror file to match the model.

**Update Model Now** Click to change the code stored in the model to match the mirror file.

Visio Enterprise lists the changes it makes to the model or mirror file.

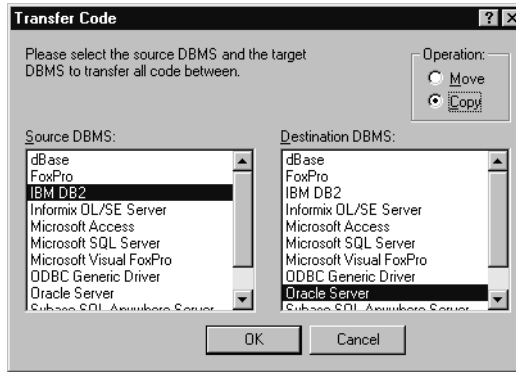
- 3 Click Refresh to update the list of code in need of synchronization. Or click Close.

## Transferring code to a new target platform

A single model that contains code can be used to generate a schema for multiple platforms. Using the Transfer command (Database > Code > Transfer), you can move your code, or a copy of it, from one platform to another. As you work in your model, you can switch between platform-specific versions of the code by changing the default driver for the model. The Code window provides access to only the version of code for the currently selected database driver. You can create as many platform-specific copies of your code as you need.

**Figure 9-6**

To enable a model to support multiple database platforms, you can transfer code from one platform to another by choosing Database > Code > Transfer.



Transferred code is platform-specific. When you use the Transfer command, Visio Enterprise moves or copies all the code in the model as text without performing any changes. You can then customize the code as necessary for the intended platform by opening it in the Code Editor.

### To transfer code:

- 1 Choose Database > Code > Transfer.  
The Transfer dialog box appears.
- 2 Under Operation, choose Move to transfer all the code to the new platform, or choose Copy to transfer a copy of the code, leaving the original implementation.
- 3 Under Source DBMS, select the database you reverse engineered. Under Target DBMS, select the new platform for which you want to generate the code.
- 4 Click OK.

When the transfer is complete, a message appears in the Info tab of the Output window.

### To switch between code implementations in your model:

- 1 Choose Database > Drivers.
- 2 In the Database Drivers dialog box, click the Drivers tab. Under Installed Visio Drivers, choose the driver for the platform to which you transferred the code.
- 3 Click OK.

The list in the Code window is updated to display the platform-specific code implementation.

## Next steps

After editing code in a model or adding new code, you can generate a new database schema based on the model, or update an existing one. Or you can synchronize your model with the database schema from which it was extracted to see what other changes may have occurred. For additional information, see the following topics.

<i>If you want to</i>	<i>See</i>
Generate a new database schema	<a href="#">“Generating a new database” in Chapter 11, “Generating a database schema”</a>
Update the schema of an existing database	<a href="#">“Running the Update Database Wizard” in Chapter 12, “Updating models and databases”</a>
Synchronize the model with an existing database schema	<a href="#">“Running the Update Model Wizard” in Chapter 12, “Updating models and databases”</a>



# Connecting to a DBMS

Using Visio® Enterprise, you can create a database model independent of its implementation on any particular database management system (DBMS). At the time you connect to your target database you can choose a DBMS-specific driver, which makes it easy to implement a database model on more than one DBMS. For example, you could prototype a database that targets one DBMS and implement it on another; or, you could extract a model from a legacy database, reuse the components, and implement the resulting model on a different DBMS.

You must specify a Visio Enterprise database driver, an associated vendor-supplied ODBC (Open Database Connectivity) driver, and a data source when you use Visio Enterprise to

- Reverse engineer an existing database using the Reverse Engineer Wizard.
- Generate a new physical database using the Generate Wizard.
- Update a model from an existing database using the Update Model Wizard.
- Update an existing database from a model using the Update Database Wizard.

This chapter explains how to set up and use the Visio Enterprise database drivers in conjunction with 32-bit ODBC drivers to connect to and exchange information about a database model with a DBMS.

## Topics in this chapter

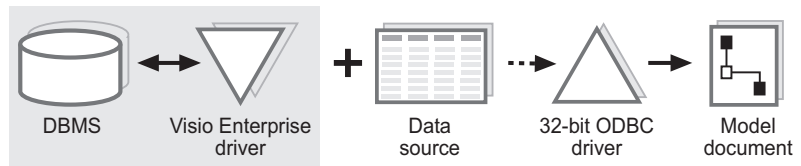
- About using database drivers..... 192
- Configuring and associating a Visio Enterprise driver ..... 194
- Creating and setting up an ODBC data source..... 196
- Connecting to Microsoft Repository ..... 198
- Next steps..... 199

# About using database drivers

A **database driver** is a file that Visio Enterprise uses to communicate with your DBMS in order to reverse engineer a database model, update a model from a database, generate a new database, and update an existing database. You use Visio Enterprise database drivers in conjunction with vendor-supplied 32-bit ODBC drivers to connect directly to your DBMS. Each driver tells Visio Enterprise the kind of DDL (Data Definition Language) script to generate, how to map data types and constraints, and how to specify extended attributes for your particular database application.

**Figure 10-1**

Visio Enterprise uses its own drivers in combination with a 32-bit ODBC driver to connect to a database.



## Enhanced database drivers in Visio Enterprise

The Visio Enterprise Setup program automatically installs the Visio Enterprise database drivers enhanced to work specifically with the Database Model template and the Microsoft ODBC Data Access Pack version 3.5 ODBC drivers, which are provided with Visio Enterprise. To determine which Visio Enterprise drivers you have installed on your system, choose Database > Drivers.

Visio Enterprise includes its own database drivers for the following DBMSs:

### Desktop database drivers

- Corel Paradox 7, 8
- Inprise (formerly Borland) dBASE III, IV, 5; Inprise Paradox 3, 4.x, 5
- Microsoft Access 2.0, 7.0, 97; Microsoft FoxPro 2.0, 2.5, 2.6; Microsoft Visual FoxPro 5.0, 6.0

### Client/server database drivers

- IBM DB2 Universal Database 2.1.2, 5.0
  - Informix Dynamic Server, with support for the Universal Data Option; INFORMIX–SE 5.0; INFORMIX–OnLine 7.x; INFORMIX–Universal Server
- NOTE** Informix Software has renamed its line of database servers. INFORMIX–OnLine is now named Informix Dynamic Server. INFORMIX–Universal Server is now named Informix Dynamic Server with Universal Data Option.
- Microsoft SQL Server 4.2, 6.0, 6.5, 7.0



- Oracle 7.x, 8
- Sybase Adaptive Server 10, 11.x (formerly Sybase System); Sybase Adaptive Server Anywhere 6.x; Sybase SQL Anywhere 5.x

#### **Microsoft Database Information Model driver**

- Microsoft DBM Repository

If you are developing a database for a DBMS other than one listed here, you can use the Visio Enterprise ODBC Generic driver with a vendor-supplied 32-bit ODBC driver to connect to your DBMS. When you use the generic driver, the information you can reverse engineer, generate, or update depends on the capabilities of the vendor-supplied 32-bit ODBC driver. The vendor-supplied driver must be:

- From an ODBC Data Access Pack greater than version 2.0.
- ODBC Level 1-compliant or greater.

You can also check the Visio Corporation Web site ([www.visio.com](http://www.visio.com)) for new and updated database drivers.

### **Before you connect**

The procedures in Part III of this book assume that you have installed and correctly configured the appropriate database software, client software if necessary, and 32-bit ODBC drivers. If you have difficulty, your system administrator can assist you with

- Obtaining access and rights to your particular desktop or SQL-based client/server database.
- Installing and configuring the appropriate client software on your local drive for a client/server database.
- Installing the correct vendor-supplied 32-bit ODBC drivers for your DBMS on your local drive.
- Configuring vendor-supplied 32-bit ODBC database drivers correctly using the Microsoft ODBC Data Source Administrator.

For details about configuring and creating a data source using the Microsoft ODBC Data Source Administrator, see that product's online help.

When you have completed these tasks, you can configure the Visio Enterprise database driver and associate it with the 32-bit ODBC driver for your specific DBMS.

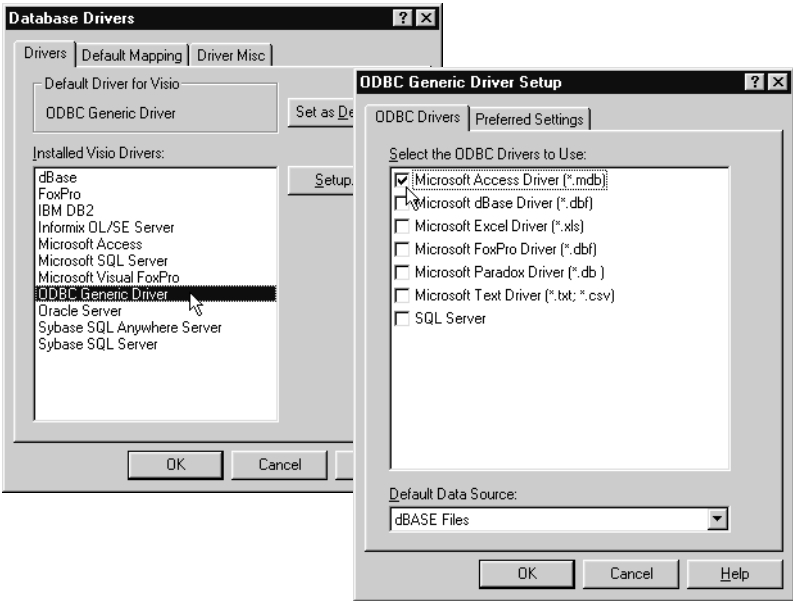
# Configuring and associating a Visio Enterprise driver

You can configure your Visio Enterprise and ODBC drivers at any time by choosing Database > Drivers. When you configure your database drivers, you specify a default driver and default mapping for column creation, associate a 32-bit ODBC driver with a particular Visio Enterprise driver, and configure default settings such as preferred settings for portable data types.

**NOTE** For an association between a Visio Enterprise driver and a 32-bit ODBC driver to work properly, the drivers you select must be designed to work with each other. For example, you would associate a Visio Enterprise Sybase driver with a vendor-supplied ODBC Sybase driver.

When you reverse engineer, generate a database, update a model, or update a database, Visio Enterprise provides a filtered list of available data sources based on the database driver associations you established.

**Figure 10-2**  
To associate a 32-bit ODBC driver with a Visio Enterprise driver, choose Database > Drivers, select an installed Visio driver, and then click Setup. On the ODBC Drivers tab, select the ODBC driver you want to use, and then click OK to close both dialog boxes.



## To configure a Visio Enterprise database driver:

- 1 Choose Database > Drivers.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 In the Database Drivers dialog box, click the Drivers tab. Under Installed Visio Drivers, select the driver you want to configure.  
*Optional* Click Set As Default to make this driver the default.
- 3 On the Default Mapping tab, under Default Category Type For Column Creation, choose the default category type for column creation and set the default data Category, Type, Size, Length, Precision, or Scale for each category type.
- 4 On the Driver Misc tab, choose default settings for DDL script generation used by the Generate and Update Database Wizards.
- 5 When you finish configuring a database driver, do one of the following:
  - Click OK to close the dialog box.
  - Click the Drivers tab, then associate the driver with a 32-bit ODBC driver by following the steps in the next procedure.

**To associate a Visio Enterprise driver with a 32-bit ODBC driver:**

- 1 Choose Database > Drivers.
- 2 In the Database Drivers dialog box, click the Drivers tab. Under Installed Visio Drivers, choose the driver with which you want to associate an ODBC driver.
- 3 Click Setup.
- 4 Click the ODBC Drivers tab, then choose the name of the 32-bit ODBC driver that you want to associate with the selected Visio Enterprise driver.

**NOTE** For an association between a Visio Enterprise driver and a 32-bit ODBC driver to work properly, the drivers you select must be designed to work with each other. For example, you would associate a Visio Enterprise Sybase driver with a vendor-supplied ODBC Sybase driver.

- 5 *Optional* Click the Preferred Settings tab, then choose default settings for portable data types and the default data source.

**NOTE** The contents of the Preferred Settings tab are driver-specific and reflect the specific features available in the DBMS. For example, the Oracle and IBM DB2 drivers have options for creating case-insensitive table and column identifiers during DDL generation.

- 6 Click OK to close the driver-specific dialog box, then click OK to close the Database Drivers dialog box.

**TIP** You can also set up or modify the association between a 32-bit ODBC driver and a Visio Enterprise driver in the Reverse Engineer Wizard, Generate Wizard, Update Model Wizard, or Update Database Wizard on the screen where you select connection options. On that screen, select an installed Visio driver, then click Setup. On the ODBC drivers tab, select the ODBC driver you want to associate with the selected Visio driver, then click OK.

# Creating and setting up an ODBC data source

After you configure and associate drivers, you can create a **data source** for a particular database. A data source includes the data you want to access and crucial information for accessing that data, such as the name of the database, the server on which it resides, and network information. Visio Enterprise refers to a data source by its **data source name (DSN)**.

**NOTE** You can create a 32-bit ODBC data source using the Microsoft ODBC Data Source Administrator, or you can create a data source as you work in the Reverse Engineer Wizard, Generate Wizard, Update Model Wizard, or Update Database Wizard. Each of these wizards contains a screen on which you can select connection options and establish ODBC DSNs.

You can edit and manage data sources you create in Visio Enterprise as you would any other data source using the ODBC Data Source Administrator. If you need help setting up, modifying, or removing an ODBC data source for your particular type of database, see the online help that accompanies the Microsoft ODBC Data Source Administrator.

## To create a data source using the ODBC Data Source Administrator:

- 1 In the Windows Control Panel, double-click the 32bit ODBC icon.
- 2 On the User DSN tab in the ODBC Data Source Administrator, click Add.

**IMPORTANT** Visio Enterprise supports only User DSNs. It does not support data sources created as File DSNs or System DSNs.

- 3 Choose the 32-bit ODBC driver you want to use for the data source, then click Finish.

The ODBC driver you select may be associated with one or more Visio Enterprise database drivers. For details about creating and editing driver associations, see [“To associate a Visio Enterprise driver with a 32-bit ODBC driver:” on page 195](#).

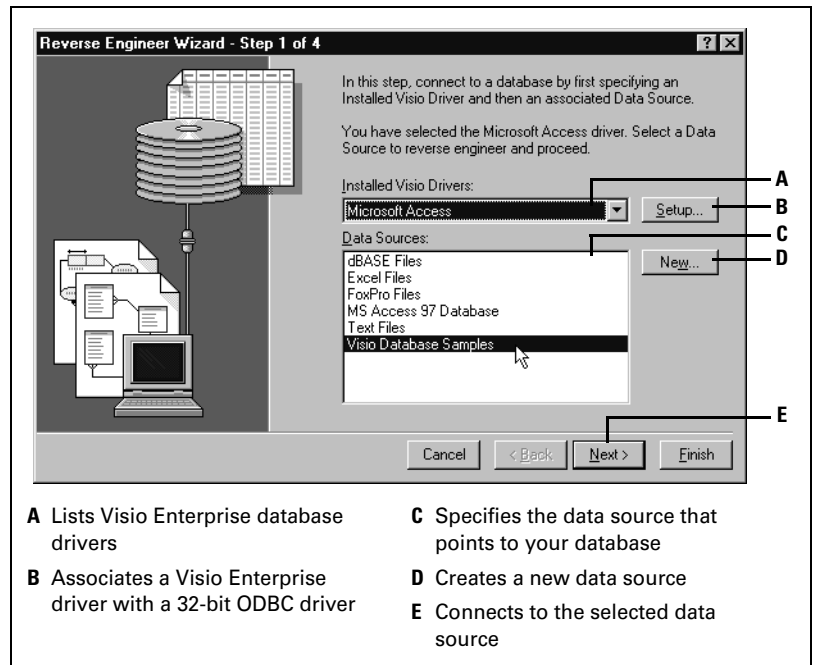
- 4 In the driver-specific dialog box that opens, type a Data Source Name and description, choose the database, then provide the required information. When you are finished, click OK.

The ODBC Data Source Administrator creates the data source.

- 5 Click OK to close the ODBC Data Source Administrator.

**Figure 10-3**

You can configure drivers and create data sources as you work in the Reverse Engineer Wizard, Generate Wizard, Update Model Wizard, or Update Database Wizard.



**To create a new ODBC data source in a Visio Enterprise wizard:**

- 1 On the wizard screen where you can select connection options, click the New button.
- 2 Select User Data Source as the type of new data source you want to create, then click Next.  
**NOTE** Visio Enterprise supports only User DSNs. It does not support data sources created as File DSNs or System DSNs.
- 3 Select the 32-bit ODBC driver you want to use, click Next, then click Finish.  
The 32-bit ODBC driver you select may be associated with one or more Visio Enterprise database drivers. For details about creating and editing driver associations, see [“To associate a Visio Enterprise driver with a 32-bit ODBC driver:” on page 195](#).
- 4 In the driver-specific dialog box that opens, type the Data Source Name, select the database, then provide the required information. When you are finished, click OK.

The ODBC Data Source Administrator creates the data source, and Visio Enterprise adds it to the list of available data sources on the screen where you select connection options.

# Connecting to Microsoft Repository

Visio Enterprise supports storage and retrieval of your database model in Microsoft Repository using the Database Information Model (DBM) open format. Using Microsoft Repository you can share database models with colleagues and use your models with other Microsoft Repository-enabled data modeling tools. Information about your database model is stored in Microsoft Repository in either a SQL Server or Microsoft Jet database, which is independent of the target implementation platform for your database.

Before you can generate, reverse engineer, or update a Visio Enterprise database model using Microsoft Repository, you must establish a connection with Microsoft Repository. You do this using the Microsoft DBM Repository driver provided with Visio Enterprise.

For details about working with Microsoft Repository in Visio Enterprise, see [“Working with Microsoft Repository” in Chapter 6, “Understanding the database modeling process.”](#)

## To connect a Visio Enterprise model to Microsoft Repository:

- 1 In a Visio Enterprise database model file, choose either Database > Update, Database > Reverse Engineer, Database > Generate, or Database > Model > Update to open a wizard.
- 2 On the connections page of the Update Database Wizard, Reverse Engineer Wizard, Generate Wizard, or Update Model Wizard, choose the DBM Repository Driver from the list of Installed Visio Drivers, then click Next.
- 3 In the Connect Microsoft Repository dialog box, choose a connection method from the options below:

**NOTE** If you connect to a Jet or SQL Server repository database, you must either use an existing repository database or create an empty repository database before you establish a connection. Microsoft Repository will not automatically create a repository database for you.

**To Jet Repository** To connect to a Jet repository database, type the following statement in the Connection String box:

```
DBQ=DatabasePath\Database.mdb
```

Replace *DatabasePath* with the directory in which the repository database resides and *Database.mdb* with the repository database file name.

**To SQL Server Repository Database** To connect to a SQL Server repository database, type the following statement in the Connection String box:

```
SERVER=ServerName;DATABASE=DatabaseName
```

**Via A Data Source Name** If you have an existing Jet or SQL Server repository database, you can connect to it using a DSN that has been configured using the ODBC Data Source Administrator. Select the DSN from the list provided.

**NOTE** For details about establishing a DSN using the ODBC Data Source Administrator, see [“Creating and setting up an ODBC data source”](#) earlier in this chapter.

- 4 If necessary, type your User Name and Password, then click OK.
- 5 Continue following the steps in the Generate Wizard, Reverse Engineer Wizard, Update Database Wizard, or Update Model Wizard.

## Next steps

Setting up database drivers and data sources are necessary steps if you want to connect to a database in order to reverse engineer a model, generate a new database, update a model from a database, or update a database from a model. For additional information, see the chapters below.

<i>If you want to</i>	<i>See</i>
Reverse engineer a database	<a href="#">Chapter 7, “Creating database models”</a>
Generate a database schema or generate a DDL script	<a href="#">Chapter 11, “Generating a database schema”</a>
Update a model from a database and reconcile any conflicts between them	<a href="#">Chapter 12, “Updating models and databases”</a>
Update or forward engineer a database from a model	<a href="#">Chapter 12, “Updating models and databases”</a>





# Generating a database schema

The process that Visio® Enterprise uses to generate a database schema for a specific database management system (DBMS) involves gathering the information from the database model, checking the model for errors, checking the model's semantics for compliance with DBMS-specific requirements, and mapping the validated model to an optimally normalized database schema.

This chapter explains how to use the Generate Wizard to generate a database schema either as a DDL (Data Definition Language) script or directly by connecting to a desktop or client/server database, or to Microsoft Repository.

For details about updating an existing database schema, see [Chapter 12, “Updating models and databases.”](#)

## Topics in this chapter

- Preparing to generate a database or a DDL script .....202
- Generating and editing a DDL script .....204
- Generating a new database .....207
- Next steps .....210

# Preparing to generate a database or a DDL script

With Visio Enterprise, you use the Generate Wizard to produce a database schema based on your model. The Generate Wizard takes you through the process of choosing options and drivers, connecting to a data source, and previewing changes. As a final step, the Generate Wizard checks your model to ensure that the information in the model does not violate any restrictions imposed by the target DBMS. If your model is free of critical errors, Visio Enterprise generates a new database, a DDL script, or both for you. If there are errors, Visio Enterprise identifies them so you can correct them in the model.

## Before you generate

Before you try to use the Generate Wizard, you should refine your model and make sure it is error-free by validating it by choosing Database > Model > Check. If your model contains errors when you choose Database > Generate to open the Generate Wizard, the Output window displays them, and the wizard will not open. For details about validation and fixing errors and warnings, see [Chapter 8, “Refining database models.”](#)

The procedures in this book assume that you have installed and correctly configured the appropriate database software and drivers. If you have difficulty, your system administrator can assist you with

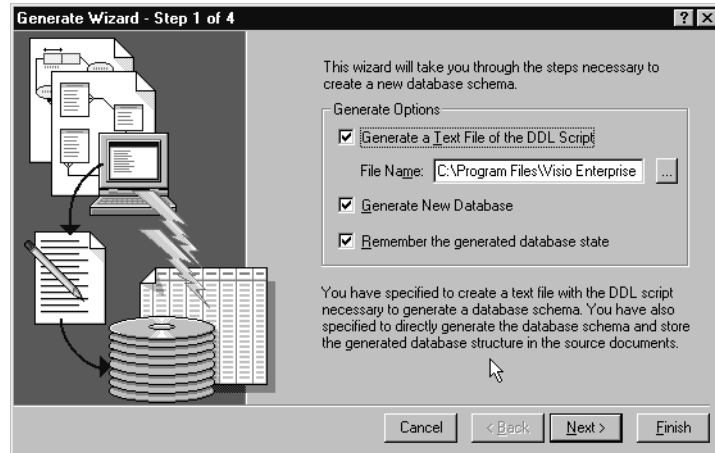
- Obtaining access and rights to your particular desktop or SQL-based client/server database.
- Installing and configuring the appropriate client software on your local drive for a client/server database.
- Installing the correct vendor-supplied 32-bit ODBC (Open Database Connectivity) drivers for your DBMS on your local drive.
- Configuring the vendor-supplied 32-bit ODBC database drivers correctly using the Microsoft 32-bit ODBC Data Source Administrator.

## Choosing a generate option for a DBMS

In the process of generating a database schema, Visio Enterprise uses enhanced database drivers in conjunction with Microsoft- or vendor-supplied ODBC drivers to connect online to your desktop or client/server database, or Microsoft Repository. The options available for generating a new database depend on the capabilities of your DBMS, and not all servers support the direct creation of a database through 32-bit ODBC drivers. Before you generate, you should determine the database generation method that works for your particular DBMS.

**Figure 11-1**

The Generate Wizard walks you step by step through creating a DDL script or a new database from your model.



In the Generate Wizard, you can generate a DDL script from your model, create a new database by connecting online to your DBMS, or both simultaneously.

**Generate a DDL script from your model.** You can preview and edit the SQL code in a DDL script, and then run the script in your database application. Visio Enterprise does not require a 32-bit ODBC data source to generate a DDL script.

**Create a new database by connecting online to your target DBMS and directly generating a database schema.** The Create Database option is available only if your server supports creation of a database through 32-bit ODBC drivers.

**IMPORTANT** The following DBMSs do not support the direct creation of a database through 32-bit ODBC drivers: all Informix DBMSs, all Oracle DBMSs, and Visual Fox Pro 3.0. When you select the drivers for these DBMSs, the Create Database option on the second screen of the Generate Wizard is dimmed.

If you are generating a database schema for a DBMS that does not support the direct creation of a new database, you can

- Generate a DDL script, which your DBMS can use to generate the database.
- Use your DBMS to create a new (empty) database on your server and then instruct Visio Enterprise to generate the database schema for it.

**Use your DBMS to create a new (empty) database on your server and use Visio Enterprise to generate a database schema for it.** You can then instruct Visio Enterprise to connect to an existing (empty) database using a data source that you prepared with your 32-bit ODBC Data Source Administrator, and then create the tables in the database.

For details about creating a new (empty) database on your server, see the documentation for your particular DBMS. For details about configuring and creating a data source using the 32-bit ODBC Data Source Administrator, see the documentation that comes with the 32-bit ODBC. For details about the Visio Enterprise database drivers, see [Chapter 10, “Connecting to a DBMS.”](#)

## Generating and editing a DDL script

If your DBMS does not support direct creation of a new database, you can use Visio Enterprise to generate a DDL script that contains the database schema for the database. You may also prefer to work with a DDL script rather than create a database directly if you want to manually review or edit the script before implementing it. Visio Enterprise does not need to connect to an ODBC data source to generate a DDL script.

On the first screen of the Generate Wizard, you can choose to generate a DDL script, a database schema, or both simultaneously.

**NOTE** This section describes generating a DDL script. The following topic, [“Generating a new database,”](#) describes generating a database schema.

DDL scripts, written in SQL code, are used to create, modify, and delete databases objects such as tables, columns, rules, and indexes. Visio Enterprise can generate a text file of the DDL script based on your model. You can then review, modify, and run the text file from your DBMS.

Once you have generated a DDL script, you can open, review, and edit the script using any text editor. You might want to review a script to verify its contents, or edit it to modify a specific component’s properties.

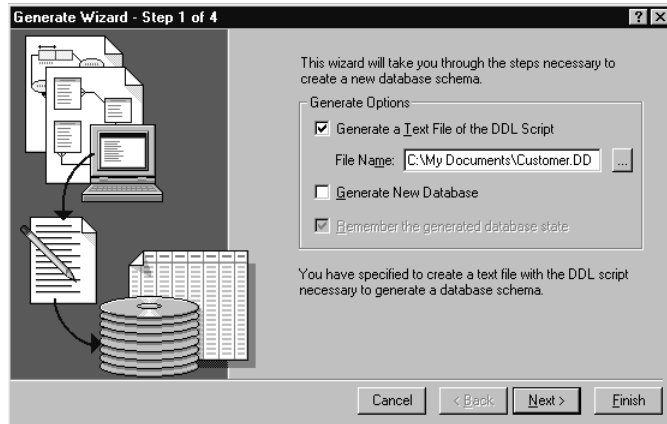
**NOTE** The Generate Wizard only creates a DDL script; it does not implement it.

### To generate a DDL script:

- 1 Choose Database > Generate to begin working in the Generate Wizard.  
**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.
- 2 On the first screen of the Generate Wizard, check Generate A Text File Of The DDL Script, enter a path and file name for the new script in the File Name box, then click Next.

**Figure 11-2**

To generate a new DDL script, choose Generate A Text File Of The DDL Script on the first screen of the Generate Wizard.

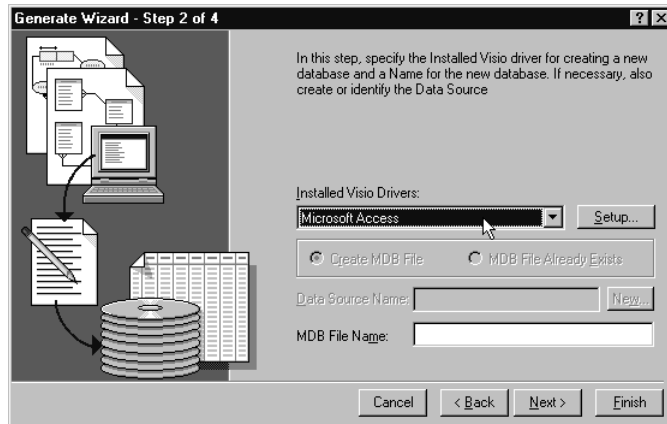


- 3 On the second screen, do the following:

From the Installed Visio Drivers list, choose a driver. If you have not already associated a Visio Enterprise driver with a particular 32-bit ODBC data source, click Setup to create the association and to specify default and preferred settings for the 32-bit ODBC driver now.

**Figure 11-3**

The second screen of the Generate Wizard contains options for selecting drivers and specifying ODBC data sources.

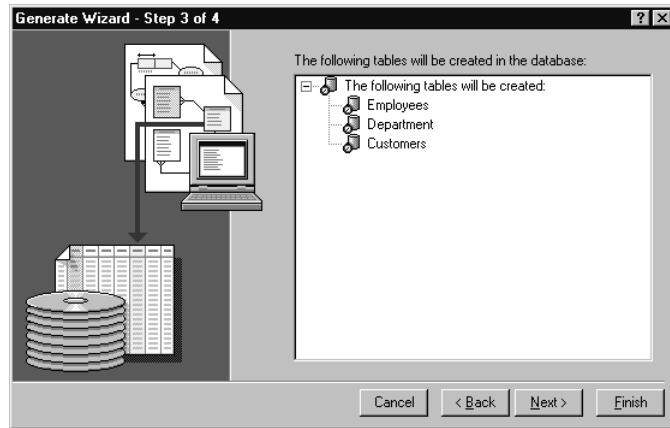


Enter a database, directory, file, or schema name, then click Next.

- 4 On the third screen, review the list of tables and views that will be created in the new database. To make any changes to the model, click Cancel. To revise options in the Generate Wizard, click Back. Click Next to continue.

**Figure 11-4**

Preview the tables that will appear in the new database in the Generate Wizard.



The Generate Wizard then validates the database schema.

- 5 On the last screen, review the Warnings box for any warnings that resulted during the validation of the database schema. The Output window lists any errors and warnings along with a brief description; double-clicking an error in the Output window takes you to the source of the error in your model.

**TIP** To see explanations and tips for correcting errors and warnings, select an error message in the Output window and press F1.

For details about how to respond to error messages, see [“Validating a database model” in Chapter 8, “Refining database models.”](#)

- 6 Click Finish to create the DDL script, or click Cancel to return to the Output window and correct any errors or warnings.

When the Generate Wizard finishes, Visio Enterprise asks if you want to view the generated DDL script. Be sure to review the script carefully before you run it in your DBMS. You can open, review, and edit the script in any text editor.

**NOTE** If you modified and then ran the DDL script that Visio Enterprise generated, you must synchronize the tables in the model with the tables in the new database. For details, see [Chapter 12, “Updating models and databases.”](#)

**To edit a DDL script immediately after you generate it:**

- 1 Generate a DDL script following the instructions in the Generate wizard. For details, see [“To generate a DDL script:” on page 205](#).
- 2 After you click Finish, you will be asked if you want to view the DDL script. Click Yes.
- 3 The DDL script opens in Notepad. You can view or edit the document.
- 4 Choose File > Save to save your changes.
- 5 Choose File > Exit to close Notepad.

**To edit a DDL script at any time:**

- From any text- or code-editing application, open the DDL script you wish to view or edit.  
The DDL script appears, and you can review or edit it.

## Generating a new database

Once your model is complete, you can use the Generate Wizard to step through the process of generating a new database schema. Whether you are generating a new database, a DDL script, or both simultaneously, choose Database > Generate, and then follow the instructions in the Generate Wizard.

**NOTE** This section describes in detail the basic tasks of generating a new database online. The previous topic describes generating a DDL script.

**To generate a database schema:**

- 1 Choose Database > Generate to begin working in the Generate Wizard.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the first screen, check Generate New Database.

*Optional* To store the generated database structure in the source document, check Remember The Generated Database State. Selecting this option creates a record of the extracted database objects, which enables the Update Database and Update Model Wizards to work correctly. For details about the Update wizards, see [“Keeping a model and database in sync” in Chapter 12, “Updating models and databases.”](#)

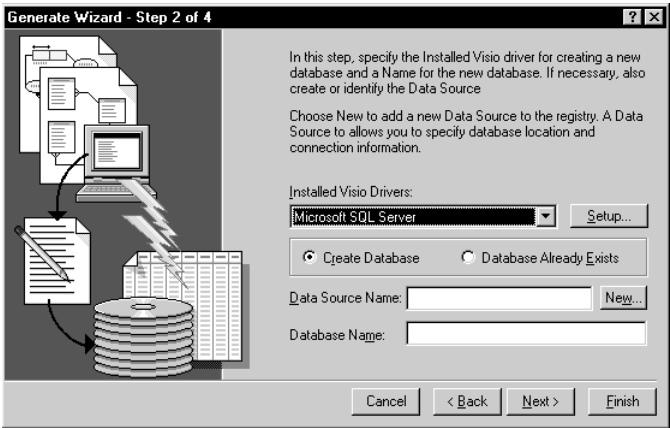
Click Next.

**Figure 11-5**  
To generate a new database schema, choose Generate New Database on the first screen of the Generate Wizard.



- 3 On the second screen, do the following:
- From the Installed Visio Drivers list, choose a driver. If you have not already associated a Visio Enterprise driver with a particular 32-bit ODBC data source, click Setup to create the association and to specify default and preferred settings for the 32-bit ODBC driver now. For details, see [“To create a stored procedure, function, or other global DDL code:” on page 181](#).
- Click Create <item> to create a new database file, or <item> Already Exists to populate an existing file with the contents of your database model. The names of the options change depending on the drivers you have selected. When you have selected a driver for a DBMS that does not support direct creation of a new database, the Create <item> option is dimmed.

**Figure 11-6**  
The second screen of the Generate Wizard contains options for selecting drivers and specifying ODBC data sources.





**Create <item> or Data Source Name** Enter a valid Data Source Name if you want to generate a database schema for an empty database you prepared ahead of time using your 32-bit ODBC Data Source Administrator. Or, to create a new, empty database as part of the generation process, click New to create a new data source, then follow the instructions on the screens.

**NOTE** If you need more information, see your documentation for the 32-bit ODBC Data Source Administrator.

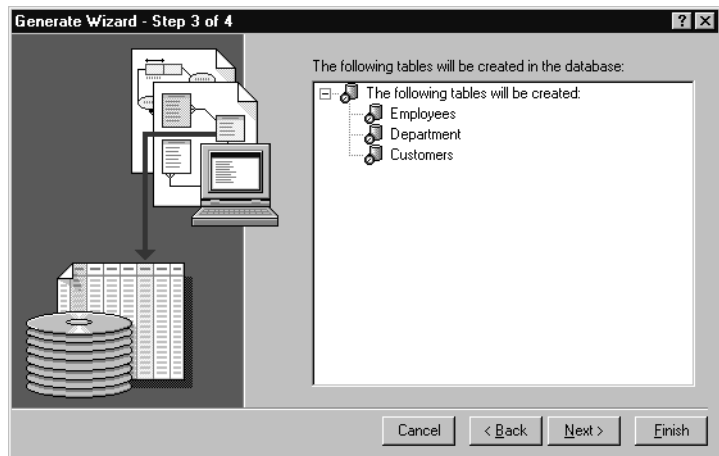
Enter a file, database, directory, or schema name, then click Next.

**<item> Already Exists** The wizard will use an existing file, database, directory, or schema. Click Next to continue.

- 4 *Optional* In the Connect Data Source dialog box, type a user name and password, then click OK.
- 5 On the next screen, review the list of tables and views that will be created in the new database. To make any changes to the model, click Cancel. To revise options in the Generate Wizard, click Back. Click Next to continue. The Generate Wizard then validates the database schema.

**Figure 11-7**

Preview the tables that will appear in the new database in the Generate Wizard.



- 6 On the last screen, review the Warnings box for any warnings that resulted during the validation of the database schema. The Output window lists any errors and warnings along with a brief description; double-clicking an error in the Output window takes you to the source of the error in your model.

**TIP** To see explanations and tips for correcting errors and warnings, select an error message in the Output window and press F1.

For details about how to respond to error messages, see [“Validating a database model” in Chapter 8, “Refining database models.”](#)

- 7 Click Finish to create the new database schema, or click Cancel to return to the Output window and correct any errors or warnings.

**NOTE** If Visio Enterprise finds any errors while generating the script, it displays them in the Output window along with a brief description of the error. For details about how to respond to error messages, see [“Validating a database model” in Chapter 8, “Refining database models.”](#)

## Next steps

You can use the DDL script or database schema you created with Visio Enterprise in your DBMS to update and maintain a database easily as your company’s information requirements change. After you have reviewed and tested the database, you can archive the project and source documents. If you need to make more changes to the database, you can review the following chapters.

<i>If you want to</i>	<i>See</i>
Synchronize your data model after generating a database from a DDL script that you modified	<a href="#">Chapter 12, “Updating models and databases”</a>
Make changes to the model and alter the database	<a href="#">Chapter 12, “Updating models and databases”</a>

# Updating models and databases

Visio® Enterprise keeps your model in sync with your database while you work and makes it easy to update the database schema when your model is complete. Whether you have created a new database model or are reverse engineering an existing database, you need a way to synchronize any changes made to the database while you make refinements in your model. Visio Enterprise uses a unique three-way comparison that tracks differences between the model and database, so that you can migrate changes from the model to the database, or from the database to the model.

To update a model from the database, use the Update Model Wizard, which provides options for resolving discrepancies. To update a database from a model, typically the final step in the reverse engineering process, use the Update Database Wizard. This chapter explains how to use these wizards to synchronize and update models and databases.

**NOTE** The Update Model and Update Database Wizards are available only when you are working in the Database Modeling solution.

## Topics in this chapter

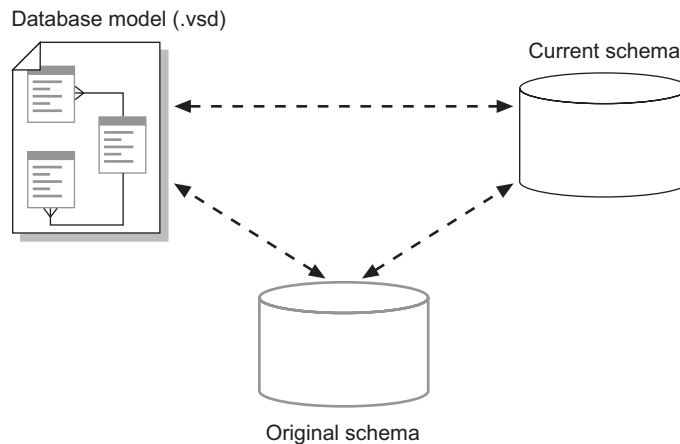
- Keeping a model and database in sync .....212
- Running the Update Model Wizard .....218
- Running the Update Database Wizard .....220
- Next steps.....225

# Keeping a model and database in sync

If the database schema has changed since the last time you generated it, you can **update** either the database or your model to keep them in sync. For example, in a large project in which several people are concurrently working in a database, someone could add or remove tables from the database after you extracted the schema on which your model is based. Using the Update Model and Update Database Wizards, you can interactively compare a model diagram to a selected database schema, and then address any items that are out of sync.

**Figure 12-1**

The automatic three-way comparison process keeps a model in sync with a database by comparing changes in the model and database to the original state of the database at the time you extracted its schema.



To synchronize a model and database, Visio Enterprise performs the following steps in a process called **three-way comparison**:

- Compares the current state of the database schema to its original state when you extracted the schema
- Compares the original state of the database schema to your model
- Compares the current state of the database schema to your model

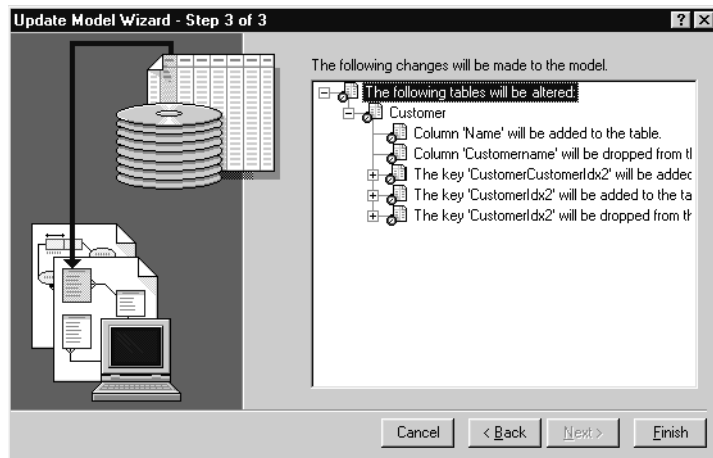
Visio Enterprise evaluates these comparisons to detect any discrepancies between the model and database schema so that you can merge changes and update one or the other as necessary. The three-way comparison process occurs automatically as a step in both the Update Model and the Update Database Wizards. For example, suppose someone deleted a table from the database since the last time you extracted its schema. When you run either the Update Model Wizard or Update Database Wizard, it detects the deleted table and lists it as a conflict. Depending on which wizard you are using, you can resolve the discrepancy in one of these ways:

- Update the model to match the current state of the database schema, an option in both the Update Model and Update Database Wizards.
- Change the database schema to match the model, an option in the Update Database Wizard.
- Return the database to its original state when you extracted the schema, an option in the Update Database Wizard.
- Ignore the conflict for now.

You can choose, item by item, how to resolve each discrepancy that Visio Enterprise uncovers. Following on the previous example, if you choose to update your model, the table is deleted from the model diagram. If you choose to update the database, the table is added back to the database.

**Figure 12-2**

Each discrepancy between a model and a database schema is listed in the wizard.



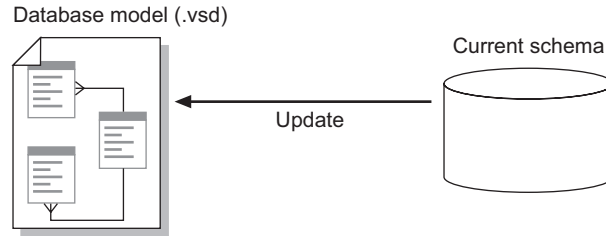
In addition, you can choose to document any conflicts or inconsistencies between the database schema and the model in a log file. You can set this option on the first screen of both wizards.

## Updating a model from a database

To ensure that your model reflects the current state of the database from which it was generated or extracted, use the Update Model Wizard (Database > Model > Update). The wizard compares the database schema to the schema represented by the model. For each difference that the wizard detects, such as duplicate or missing tables, you can choose whether to update the model to reflect that change.

**Figure 12-3**

The Update Model Wizard makes changes to a model to synchronize it with the current state of a database schema.



You should update a model in any of these cases:

- You want to detect any changes that may have occurred in a database and incorporate them into your model.
- You used the Generate Wizard to create a DDL (Data Definition Language) script, which you executed to alter your database, and now want to update the model in Visio Enterprise.

The following table describes the differences between each object in a database schema and your model that can be detected by the Update Model Wizard.

### Differences between a database schema and model detected by the Update Model Wizard

<i>Database object</i>	<i>Type of schema change detected</i>
Table	Deleting; changing its check clause, extended attribute, type (for typed tables)
Column	Adding or deleting; changing its name, data type, collection type, check clause, or extended attributes
View	Deleting or changing its structure
Primary key	Adding or deleting; changing its name or structure
Relationship (foreign key)	Adding or deleting; changing its name or structure (columns, cardinality, or referential integrity)
Index	Adding or deleting; changing its columns, collation, or uniqueness

### Differences between a database schema and model detected by the Update Model Wizard (*continued*)

<i>Database object</i>	<i>Type of schema change detected</i>
Composite type	Deleting; changing its structure (named, distinct, domain) or alias (for collection type or aliased data type)
Field in a type	Adding or deleting; changing its name, structure (data types or, for object-relational, collection type), check clause, or extended attributes
Code	Deleting; changing code statements or code type (stored procedure, function, trigger, etc.)

## Updating a database from the model

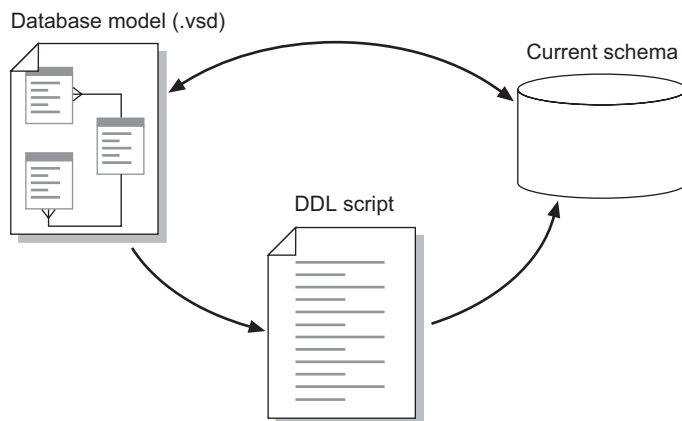
When you use the Update Database Wizard (Database > Update), you can change the structure of a database to match the contents of your model. The Update Database Wizard calculates the net changes to a database schema based on your model, prompts you to resolve any conflicts, and then generates a DDL script or directly alters the database.

You should update a database in any of these cases:

- You reverse engineered the database in Visio Enterprise, and now want to forward engineer the changes in your model to the database.
- After you reverse engineered the database, changes were made to the original database, and now you want to update your model.
- You created a new model from scratch that you want to insert into an existing database.

**Figure 12-4**

The Update Database Wizard makes changes directly to a database or creates a DDL script that you can execute to make changes based on the refinements you have made to a model.



The Update Database Wizard (Database > Update) provides two ways to update the database, directly or through a DDL script.

- The wizard connects to your target database and directly alters an existing database schema or creates a new one.
- The wizard creates a DDL script of the changes that you want to make to the database. You can then execute the script from your database application.

If the wizard cannot directly generate update statements for the target database, it may also create a DML (Data Manipulation Language) script to add any data from the existing tables to the newly created ones.

The following table describes the differences between each object in a database schema and your model that can be detected by the Update Database Wizard.

**Differences between a database schema and model detected by the Update Database Wizard**

<i>Database object</i>	<i>Type of change detected</i>
Table	Deleting from model or database; adding to model; changing its name in model; changing its check clause, extended attribute, type (for typed tables) in model or database
Column	Adding to or deleting from model or database; changing its name, data type, collection type, check clause, or extended attributes
View	Deleting from model or database; adding to or changing name in model; changing its structure
Primary key	Adding to or deleting from model or database; changing its name or structure
Relationship (foreign key)	Adding to or deleting from model or database; changing its name or structure (columns, cardinality, or referential integrity)
Index	Adding to or deleting from model or database; changing its columns or uniqueness in model or database; changing collation in database
Composite type	Deleting from model or database; adding to or changing name in model; changing its structure (named, distinct, domain) or alias (collection type or aliased data type) in model or database



### Differences between a database schema and model detected by the Update Database Wizard (*continued*)

<i>Database object</i>	<i>Type of change detected</i>
Field in a type	Adding to or deleting from a model or database; changing its name, structure (data types or, for object-relational, collection type), check clause, or extended attributes
Code	Deleting from model or database; adding to or changing the name in model; changing the code or its type (stored procedure, function, trigger, etc.)

**NOTE** You can also use the Update Database Wizard to update your model based on a database schema, just as you can with the Update Model Wizard. The key difference between these wizards is that only the Update Database Wizard can make changes to a database.

## Connecting to a data source

As part of updating a model or database, you must specify how to locate and communicate with the database with which you want to work. Whether you are using a desktop or client/server database, or Microsoft Repository, you can use the Update Model Wizard or Update Database Wizard to

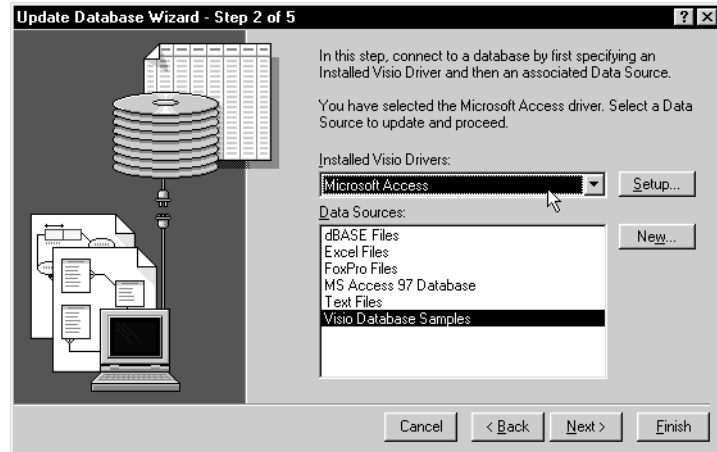
- Select an installed Visio Enterprise database driver.
- Associate a 32-bit ODBC (Open Database Connectivity) driver with an installed Visio Enterprise database driver.
- Select or create a 32-bit ODBC data source. For details about creating a data source, see the documentation for the Microsoft 32-bit ODBC Data Source Administrator.

You can also set up the connections before working with the wizards. For details about performing these steps, see [Chapter 10, “Connecting to a DBMS.”](#) If you are updating a database or model stored in Microsoft Repository, you must connect with Microsoft Repository by using the Microsoft DBM Repository driver provided with Visio Enterprise. For details, see [“Connecting to Microsoft Repository” in Chapter 10, “Connecting to a DBMS.”](#)

When you generate a DDL script that performs the updates, an ODBC data source is not required unless your target database requires server information in its CREATE DATABASE statement. The Update Database Wizard detects whether the database requires an ODBC data source and provides options for you to choose.

**Figure 12-5**

You can select and associate the drivers required to connect with your target data source in the Update Database and Update Model Wizards.



## Running the Update Model Wizard

The Update Model Wizard steps you through the process of connecting Visio Enterprise to your target database, comparing the model diagram to the actual database's schema, and then resolving any conflicts. If the wizard detects differences between the model and database, you can choose to update the model or ignore the conflict for now.

### To synchronize a database with your model:

- 1 Choose Database > Model > Update.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the first screen, do the following:

From the Installed Drivers list, choose the driver. If you have not already associated a Visio Enterprise database driver with a particular 32-bit ODBC data source, click Setup to create the association now. For details, see [“To associate a Visio Enterprise driver with a 32-bit ODBC driver:” on page 195.](#)

Under Data Sources, choose the source of the database you are updating, then click Next. If you have not already created a data source for the existing database, click New to create one, then follow the instructions on the screens. For more details, see [“To create a new ODBC data source in a Visio Enterprise wizard:” on page 197](#). When you finish creating a new data source, click OK.

Visio Enterprise adds its name to the Data Sources list.

- 3 In the Connect Data Source dialog box, type a user name and password, then click OK.
- 4 On the next screen, select an item in the list of conflicts, then click the resolution option you want to use:

**No Change** Do not change either the model or the database.

**Update Model** Change the model to match the database schema.

Repeat for each item in the list. When you are finished, click Next.

- 5 On the next screen, review the changes that the wizard will make, then click Next.
- 6 On the final screen, click Finish to accept the changes and update the model.

The wizard checks the model for semantic and implementation errors and writes a list of progress notes, errors, and warnings in the Output window. If validation succeeds, the wizard updates the model.

If validation fails, you can correct the errors, then run the Update Model Wizard again. For example, if a table’s primary key has been removed in the database, validation fails. For details about locating and correcting errors, see [“Validating a database model” in Chapter 8, “Refining database models.”](#)

# Running the Update Database Wizard

The Update Database Wizard walks you step by step through the process of connecting to a database, comparing the model in Visio Enterprise to the target database, and then resolving any differences by updating the model or the database. To update the database, you can choose to generate a DDL script that performs the changes or directly alter the database.

Before you update the database schema, you should

- Verify that your model is valid by choosing Database > Model > Check. The Update Database Wizard prompts you to check your model if it detects errors.

For details about validating models, see [“Validating a database model”](#) in Chapter 8, [“Refining database models.”](#)

- Verify that your system is set up correctly with appropriate access rights to the target desktop or client/server database and all the necessary drivers and client software required by your target database.

For details, see [Chapter 10, “Connecting to a DBMS.”](#)

**NOTE** Before you run the Update Database Wizard to alter any database, be sure to use your usual safety measures to protect your data.

## Creating a script that updates a database

With the Update Database Wizard, you can generate a DDL script for updating and physically altering a database based on a database model diagram. When the wizard connects to the target database, it uses the server name, database name, and other information to generate the appropriate DDL (and, if necessary, DML) statements in a text file. The script’s exact contents depend on the target database driver, which tells the wizard the type of script to create, the data types that are supported, and other options specific to the database. You can then review the script, edit it if you want, and execute it later from within your database application.

The Update Database Wizard generates a text file of DDL statements to create, modify, and delete databases objects, such as tables, columns, and indexes, based on your model. To control how the wizard generates comments in the DDL script it creates, choose Database > Drivers, then click the Driver Misc. tab. On this tab you can check options for how the wizard creates the SQL statements in the update script.

**TIP** After you generate a DDL script, open it in any text editor to review it for errors before running the script in your database application.

### To create an update script:

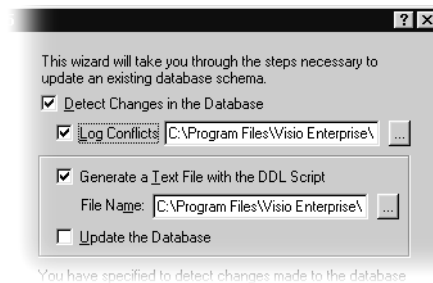
- 1 Choose Database > Update.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the first screen, do the following:
  - Check Detect Changes In The Database.
  - To write discrepancies to a file, check Log Conflicts, then specify a file name and path.
  - Check Generate A Text File With The DDL script, then specify a file name and path for the script.

**Figure 12-6**

On the first screen of the Update Database Wizard, you can choose to create a script to alter your database and create a log file of changes.



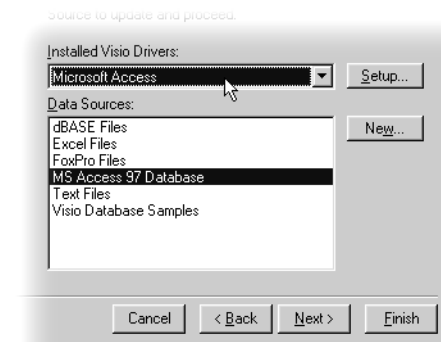
Click Next.

- 3 On the second screen, do the following:

From the Installed Visio Drivers list, choose a driver. For example, if you want to update an existing Oracle database, select the Oracle driver. If you have not already associated a Visio Enterprise driver with a particular 32-bit ODBC data source, click Setup to create the association now.

**Figure 12-7**

Choose the Visio Enterprise driver associated with the database you want to update.



Under Data Sources, choose the source of the database you are updating, then click Next. If you have not already created a data source for the existing database, click New to create one, then follow the instructions on the screens. For more details, see [“To create a new ODBC data source in a Visio Enterprise wizard:” on page 197](#). When you finish creating a new data source, click OK.

Visio Enterprise adds its name to the Data Sources list.

- 4 In the Connect Data Source dialog box, type a user name and password, then click OK.
- 5 On the next screen, select an item in the list of conflicts, then click the resolution option you want to use:  
**No Change** Do not change either the model or the database.  
**Update Model** Change the model to match the database.  
**Update Database** Change the database to match the model.
- 6 On the next screen, any updates to the model that you specified are listed. Review the changes, then click Next.  
  
For example, if your model contains a table that is not in the database, and you instructed the wizard to update the model, the table will be deleted from the model.
- 7 On the next screen, any updates to the database schema that you specified are listed. Review the changes, then click Next.  
  
For example, if your model contains a table that is not in the database, and you instructed the wizard to update the database, the table will be added to the database.
- 8 On the final screen, click Finish to accept the changes and create an update script.

The wizard checks the model for semantic and implementation validity and writes a list of progress notes, errors, and warnings in the Output window.

- If validation succeeds, the wizard prompts you to save the script in the file you defined on the first screen of the wizard.
- If validation fails, click Cancel to exit the wizard. For example, if a table's primary key has been removed in the database, validation fails. Correct the errors before running the Update Database Wizard again.

For details about locating and correcting validation errors, see [“Validating a database model” in Chapter 8, “Refining database models.”](#)

**NOTE** The Update Database Wizard only creates a DDL script; it does not implement it. You must run the script manually from within your database application.

## Updating a database schema directly

Rather than generate a DDL script, the Update Database Wizard can directly connect to a target database and update it, altering existing table definitions and adding and removing tables. You can also update the schema directly and at the same time generate a DDL script.

### To update a database schema directly:

- 1 Choose Database > Update.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 On the first screen, do the following:

- Check Detect Changes In The Database.
- To write discrepancies to a file, check Log Conflicts, then specify a file name and path.
- Check Update The Database.
- To create a DDL script as well, check Generate a Text File With The DDL Script, then enter a file name.

Click Next.

- 3 On the second screen, do the following:

From the Installed Visio Drivers list, choose a driver. For example, if you want to update an existing Oracle database, select the Oracle driver. If you have not already associated a Visio Enterprise driver with a particular 32-bit ODBC data source, click Setup to create the association now.

Under Data Sources, choose the source of the database you are updating, then click Next. If you have not already created a data source for the existing database, click New to create one, then follow the instructions on the screens. For more details, see [“To create a new ODBC data source in a Visio Enterprise wizard:” on page 197](#). When you finish creating a new data source, click OK.

Visio Enterprise adds its name to the Data Sources list.

- 4 In the Connect Data Source dialog box, type a user name and password, then click OK.

- 5 On the next screen, select an item in the list of conflicts, then click the resolution option you want to use:

**No Change** Do not change either the model or the database.

**Update Model** Change the model to match the database.

**Update Database** Change the database to match the model.

Repeat for each item, then click Next.

- 6 On the next screen, any updates to the model that you specified are listed. Review the changes, then click Next.

For example, if your model contains a table that is not in the database, and you instructed the wizard to update the model, the table will be deleted from the model.

- 7 On the next screen, any updates to the database schema that you specified are listed. Review the changes, then click Next.

For example, if your model contains a table that is not in the database, and you instruct the wizard to update the database, the table is added to the database.

- 8 On the final screen, click Finish to accept the changes and update the database schema.

The wizard checks the model for semantic and implementation errors and writes a list of progress notes, errors, and warnings in the Output window.

- If validation succeeds, the wizard connects to your database and updates it with the changes you specified. A message in the Output window notifies you when the process is finished.
- If validation fails, click Cancel to exit the wizard. For example, if a table's primary key has been removed in the database, validation fails. Correct the errors in the model before running the Update Database Wizard again.

For details about locating and correcting errors, see [“Validating a database model” in Chapter 8](#), [“Refining database models.”](#)



# Next steps

After you update a model with the Update Model Wizard, your next step is to review and refine the model further, and to check it for errors.

After you update a database with the Update Database Wizard, your next step is to review and test the database. If you are satisfied with the changes, archive your model diagram. Otherwise, refine your model further, and then try updating again.

<i>If you want to</i>	<i>See</i>
Refine a database model and check it for errors	<a href="#">Chapter 8, “Refining database models”</a>
Generate a new database schema after synchronizing or updating	<a href="#">Chapter 11, “Generating a database schema”</a>
Update an existing database schema after updating a model	<a href="#">“Running the Update Database Wizard”</a> earlier in this chapter



# Creating reports

You can create customizable reports in Visio® Enterprise that describe your database model by displaying data definitions and formatting information. For example, you can create a report that documents your database design to present to a client. You can create reports that summarize file and database statistics or that document the tables in your model. In addition, if you are creating an object-relational model, you can also report on data types.

This chapter describes how to use the Report Wizard and the Report window to customize the information you want to appear in a report, and then to preview, print, or export that information.

## Topics in this chapter

- Generating reports ..... 228
- Summarizing file and database statistics in a report ..... 234
- Listing database tables in a report ..... 236
- Listing data types in a report ..... 238
- Next steps ..... 239

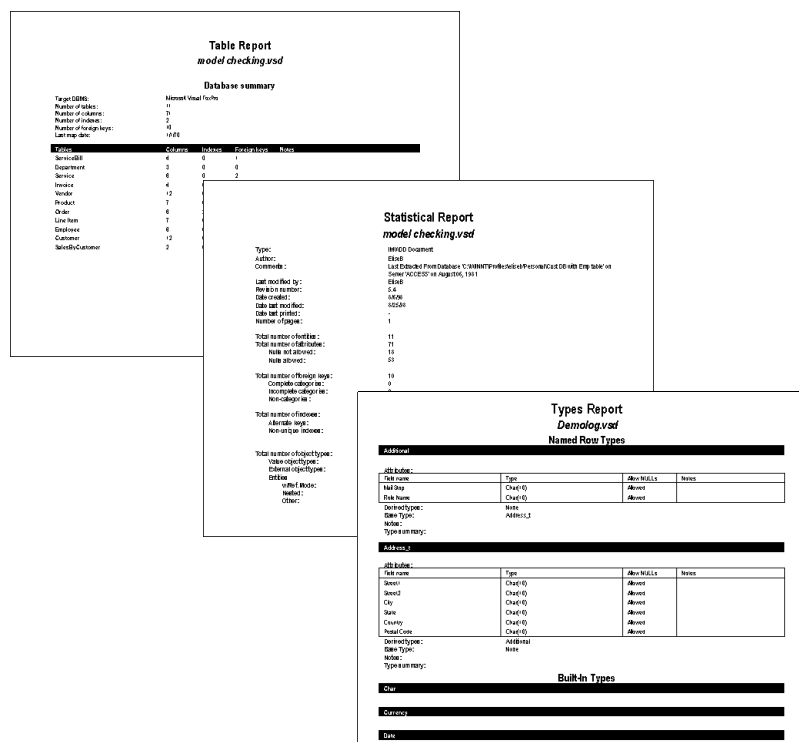
## Generating reports

You can create, print, preview, and export comprehensive reports based on the contents of the current model on the drawing page. Visio Enterprise includes three types of preformatted reports, which you can customize to include the data definitions and formats that you want. Creating a new report is a two-step process:

- 1 Use the Report Wizard to select the type of report you want to create.
- 2 Define and customize the report in the Report window.

The Report Wizard automatically launches the Report window, where you can choose options for refining the report's contents as well as printing, previewing, and exporting the report.

**Figure 13-1**  
Visio Enterprise includes three types of preformatted reports.



**To create a new report:**

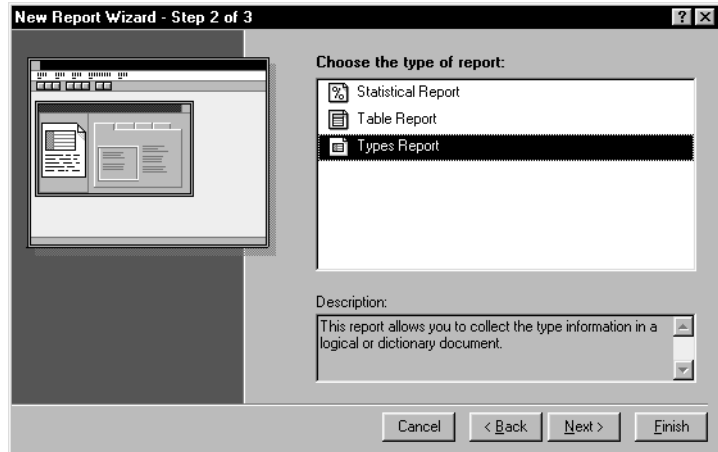
- 1** Choose Database > Report, then click Next.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 Choose the type of report you want to create, then click Finish.

**Figure 13-2**

The Report Wizard automatically generates reports summarizing model and file statistics, data types, and tables.

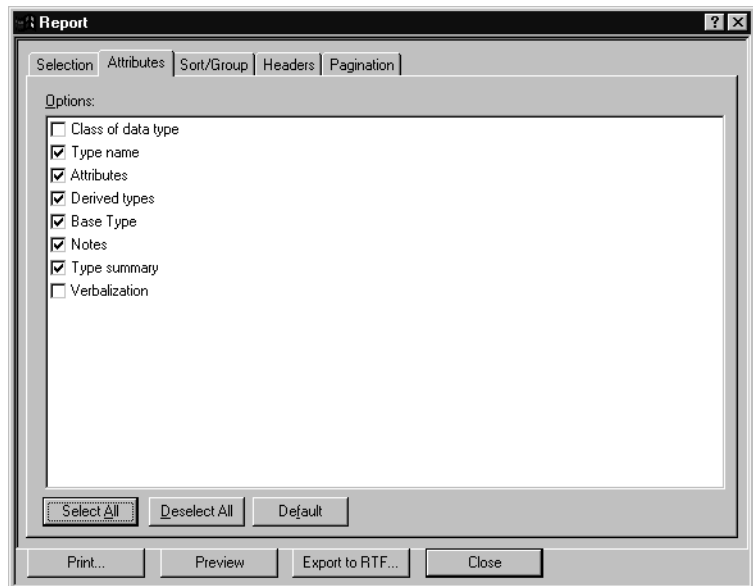


- 3 In the Report window, define and format the information you want to include in your report.
- 4 Click Print, Preview, or Export To RTF to generate the report.
- 5 Click Close to close the Report window.

For details about specifying report options, see the next section.

**Figure 13-3**

The Report window is a workspace for setting and modifying a report's definition. Depending on your report type, you can sort, group, customize, and organize the data that appears in your report.

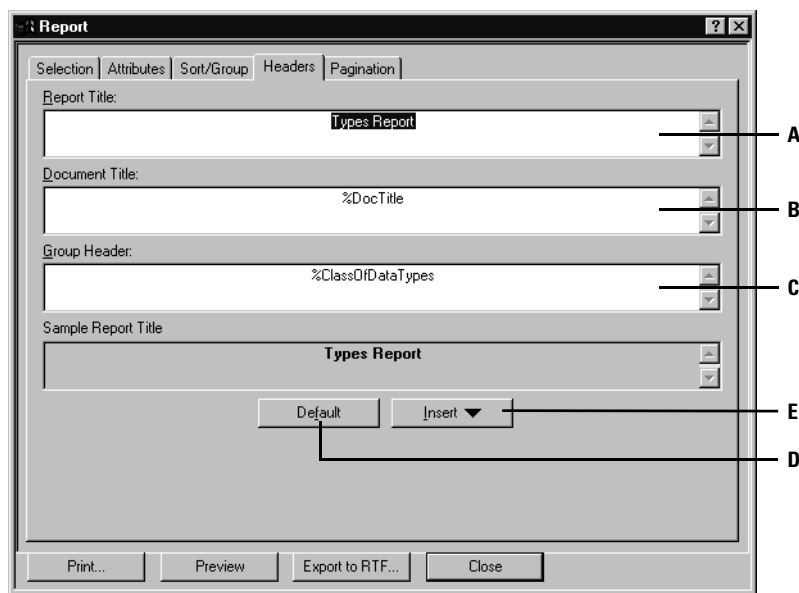


## Formatting report titles and pages

Reports in Visio Enterprise are preformatted, so that you can concentrate on quickly getting the information you need. However, you can add a title and other text to your report in a header and format the different sections of your report to help visually organize the information. When you work in the Report window, you can customize these formatting options on the Headers and Pagination tabs.

On the Headers tab, you can specify a report title, which appears at the top of the first report page, a document title, which appears beneath the report title on the first page, and a group header supplied by Visio Enterprise that separates groups of records in your report. Each group is listed on the Sort/Group tab.

**Figure 13-4** Formatting options on the Headers tab in the Report window



**A** Specifies the report title, which appears at the top of the first report page.

**B** Specifies a name for the document or a subtitle, such as the model name, which appears beneath the report title.

**C** Shows the subtitle supplied by Visio Enterprise that separates each group of records. Groups are set on the Sort/Group tab.

**D** Restores the header text to the Visio Enterprise default titles.

**E** Lists the formatting codes.

To automatically generate common header elements, such as page number, time, and date, you can use a formatting code. A formatting code is simply a percent sign (%) followed by a word.

**Formatting codes to use in a report’s header text**

<i>Code</i>	<i>Description</i>
%Author	Name specified in the Properties dialog box (File > Properties)
%Date	Current system date
%DocTitle	File name of the model diagram
%DocTitleLong	Title of the model diagram, set on the Summary tab in the Properties dialog box (File > Properties)
%File	Path and file name of the model diagram

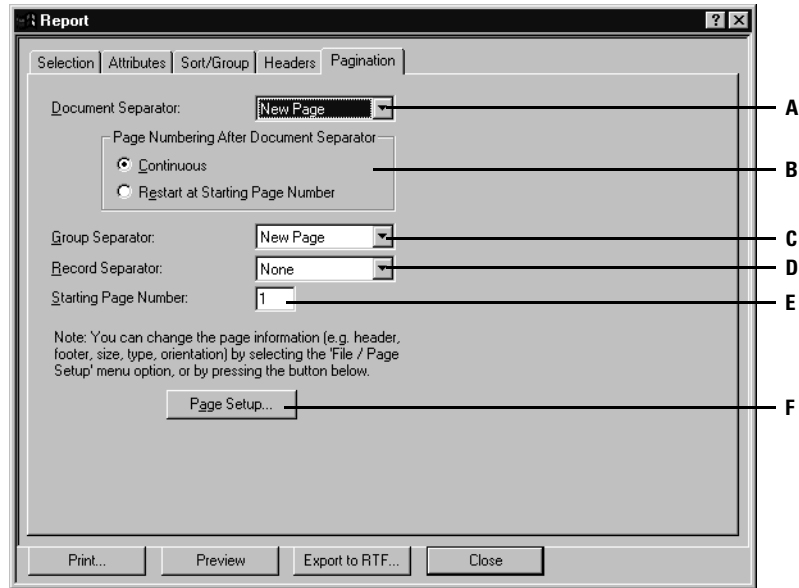
The following table describes how to work in the Headers tab.

**Adding header text**

<i>To</i>	<i>Do this</i>
Type header text	On the Headers tab, click in an edit box, then type the text you want.
Insert a formatting code in the header text	On the Headers tab, click in an edit box, click Insert, then choose a code.
Add line breaks to text	Press Ctrl+Enter.
Delete text	Select the text, then press the Delete key.

To format groups of information in your report visually, use the Pagination tab in the Report window. You can set page breaks or specify to print **separators**, which are bars that appear between records and groups of records. In a report, **record** refers to all the data about one object type; that is, entities or tables. A **group** is a set of records sorted by a key that you specify on the Sort/Group tab.

**Figure 13-5** Formatting options on the Pagination tab in the Report window



- A** If the report contains more than one page, specifies how to separate each.
- B** If you set a Document Separator, specifies to number all pages consecutively (Continuous) or to renumber after the separator with the value set for Starting Page Number.
- C** Specifies how to separate groups in the report. Groups are set on the Sort/Group tab.
- D** Specifies how to separate all the data about one entity or table in the report.
- E** Sets the first page number.
- F** Opens the Page Setup dialog box, where you can set print options.

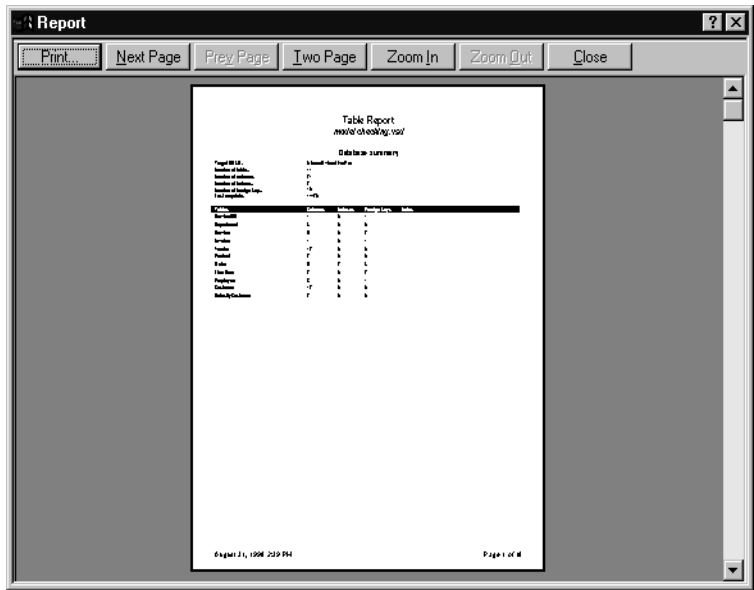


## Previewing a report

To preview a report before you print or export it, click the Preview button in the Report window. Previewing provides a view of the report so that you can verify its contents. To change the report's contents, click Close to return to the tabs in the Report window.

**Figure 13-6**

In the Report window, you can preview a report to verify its contents before you print or export it. Use the buttons to zoom the view and navigate between pages.



The following table describes how to change the view of your report as you work in the preview window.

### Changing the view of a report

<i>To</i>	<i>Click</i>
Navigate between pages	Next Page or Prev Page
Change the magnification of the view	Zoom In or Zoom Out
Change the number of pages being previewed	One Page or Two Page

## Exporting a report

To reuse the contents of a report in another application, or to customize its contents further, you can export the report as a Rich Text Format (RTF) file. Visio Enterprise exports graphic elements in a report as bitmaps and exports example data in a tabular format as part of the report. You can then open and edit the RTF file in any application that supports RTF, such as Microsoft Word.

### To export a report:

- 1 In the Report window, click Export To RTF.
- 2 In the Export To RTF dialog box, type a file name and path, then click Save.  
Visio Enterprise exports the report as an RTF file and displays a progress indicator in the status bar.
- 3 Click Close to close the Report window.

## Summarizing file and database statistics in a report

When you want a high-level summary of the contents of a database model diagram, use the Report Wizard to create a **statistical report**. In a statistical report, Visio Enterprise tallies the number and type of constraints, foreign keys, and attributes, and summarizes source file information for your model.

When you create a statistical report, you can choose to describe your model using entity relationship terms, such as entities, attributes, and relationships, or object-relational terms, such as tables, columns, and keys. You can set this option on the Preferences tab in the Report window.

**Figure 13-7**

A statistical report summarizes source file information for your model.

Statistical Report	
<i>model checking.vsd</i>	
Type:	IMUDD Document
Author:	ElseB
Comments:	Last Extracted From Database C:\WINNT\Profiles\ElseB\Personal\Cast DB with Emp table on Server ACCESS on August 06, 1991
Last modified by:	ElseB
Revision number:	5.4
Date created:	8/6/96
Date last modified:	8/25/96
Date last printed:	-
Number of pages:	1
Total number of entities:	11
Total number of attributes:	71
Nulls not allowed:	18
Nulls allowed:	53
Total number of foreign keys:	10
Complete categorize:	0
Incomplete categorize:	0
Non-categorize:	10
Total number of indexes:	2
Alternate keys:	0
Non-unique indexes:	2
Total number of object types:	70
Value object type:	70
External object type:	0
Entity:	0
with f. mode:	0
Nested:	0
Other:	0

**To create a statistical report:**

- 1 Choose Database > Report, then click Next.  
**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.
- 2 In the Report Wizard, choose Statistical Report, then click Finish.  
The Report window opens.
- 3 On the Report window tabs, do the following:  
**Preferences tab** Choose the notation you want to use in the report.  
**Headers tab** Create a title for the report.  
**Pagination tab** Set page breaks and other formatting options.
- 4 When you are finished, click Preview to see your report. Or click Print or Export To RTF. Click Close to close the Report window.

## Listing database tables in a report

With the Report Wizard, you can create a **table report** that shows how objects in your model, such as tables and columns, will appear in the database you generate or update from the model. Visio Enterprise includes two types of table reports.

- A **conceptual table report**, which describes your model using the conceptual names of objects specified in the various Properties dialog boxes.
- A **physical table report**, which describes the model using the physical names of objects specified in the various Properties dialog boxes.

To choose the type of report you want, use the Preferences tab in the Report window, where you can also choose whether to describe the model using entity relationship or object-relational terminology.

**Figure 13-8**

A physical table report itemizes the tables in your model and provides detailed information about each table's contents.

Table Report				
model checking.vsd				
Database summary				
Target DBMS:	Microsoft Visual FoxPro			
Number of tables:	11			
Number of columns:	71			
Number of indexes:	2			
Number of foreign keys:	10			
Last map date:	10/00			
Tables	Columns	Indexes	Foreign keys	Notes
ServiceBill	4	0	1	
Department	3	0	0	
Service	6	0	2	
Invoice	4	0	1	
Vendor	12	0	0	
Product	7	0	0	
Order	6	2	3	
Line Item	7	0	2	
Employee	6	0	1	
Customer	12	0	0	
SalesByCustomer	2	0	0	

A table report can also describe the attributes associated with the tables, columns, indexes, and foreign keys in your model. On the Attributes tab in the Report window, you can choose from a number of attributes in each of these categories to include in your report. If you don't want to report on a particular category, such as Foreign Key, uncheck all the attributes associated with that category.

You can sort the information that appears in your report using the Sort/Group tab in the Report window.

**To create a table report:**

- 1 Choose Database > Report, then click Next.

**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.

- 2 In the Report Wizard, choose Table Report, then click Finish.

The Report window opens.

- 3 On the Report window tabs, do the following:

**Preferences tab** Choose the type of report you want, conceptual or physical, and the type of notation to use.

**Attributes tab** Choose the database, table, column, index, and foreign key attributes you want in your report.

**Sort/Group tab** Choose the report objects that will be used to sort the report contents.

**Headers tab** Create a title for the report.

**Pagination tab** Set page breaks and other formatting options.

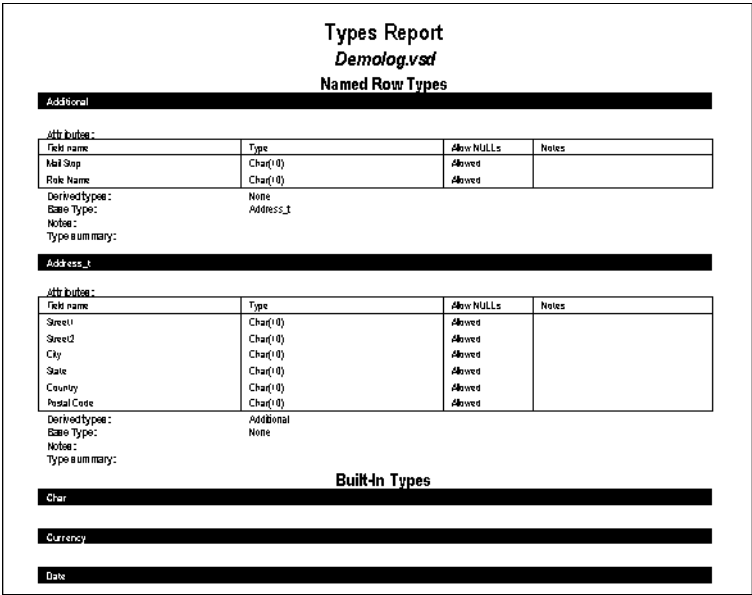
- 4 When you are finished, click Preview to see your report. Or click Print or Export To RTF. Click Close to close the Report window.

# Listing data types in a report

You can create a report of the composite, user-defined, and built-in types represented in your database model diagram. With the Report Wizard, you can create a **types report**, which displays data types in your report. Then in the Report window, you can tailor your report by choosing the data types to include.

Your types report can also include attribute information, such as type names, base types, and derived types. Choose from the options on the Attributes tab in the Report window to indicate which attributes to include. Your report is sorted by the class of data type and then by type name. To change the sorting criteria, use the options on the Sort/Group tab.

**Figure 13-9**  
A types report for a SQL3 model



### To create a types report:

- 1 Choose Database > Report, then click Next.  
**NOTE** This command is available only when you start a model in Visio Enterprise using the Database Model template.
- 2 In the Report wizard, choose Types Report, then click Finish.  
The Report window opens.
- 3 On the Report window tabs, do the following:  
**Selection tab** Choose the types you want to show in your report.  
**Attributes tab** Choose the attributes you want in your report.  
**Sort/Group tab** Choose the report objects that will be used to sort the report contents. Only the options you checked on the Attributes tab appear on this tab as records you can sort or group.  
**Headers tab** Create a report title.  
**Pagination tab** Set page breaks and other formatting options.
- 4 When you are finished, click Preview to see your report. Or click Print or Export To RTF. Click Close to close the Report window.

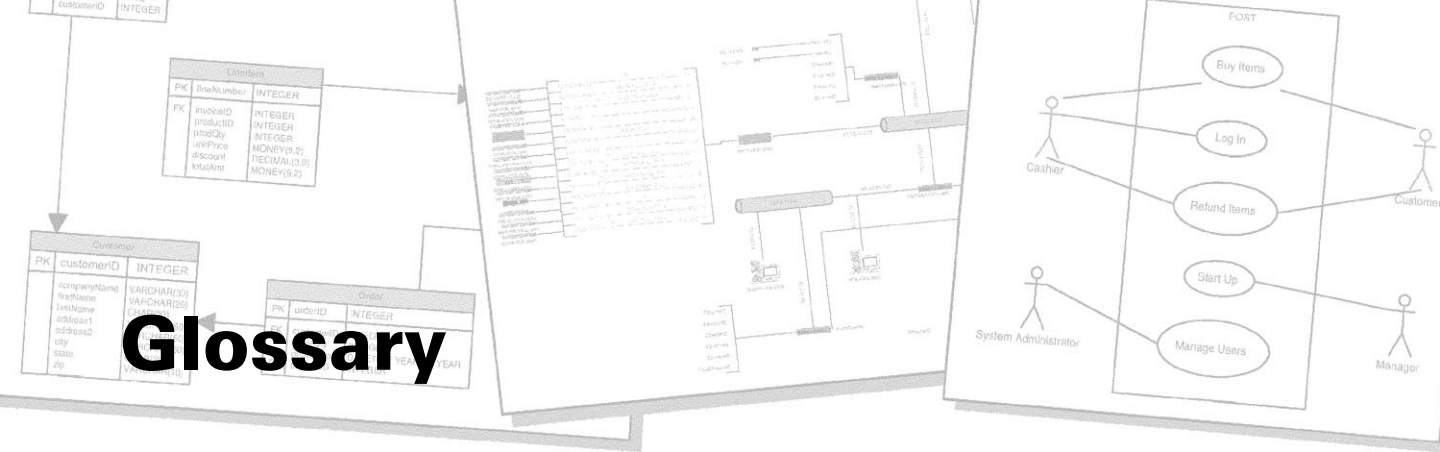
## Next steps

You can create reports at any time during the modeling process to communicate with clients, coordinate your efforts with those of coworkers, or simply to provide another view of your model's contents. Depending on where you are in the modeling or reverse engineering process, your next step after creating a report could be to refine your model further, check it for errors, or synchronize the model with the target database.

<i>If you want to</i>	<i>See</i>
Refine your model and check it for errors	<a href="#">Chapter 8, "Refining database models"</a>
Synchronize your model with the target database	<a href="#">"Updating a model from a database" in Chapter 12, "Updating models and databases"</a>
Update an existing database schema based on your model	<a href="#">"Updating a database from the model" in Chapter 12, "Updating models and databases"</a>







# Glossary

**alter** To use the Update Database Wizard in the Database Model solution to make changes directly to a database schema. *See also* [database schema](#), [updating](#).

**ARP (Address Resolution Protocol)** A protocol for mapping IP addresses to Ethernet addresses. *See also* [IP \(Internet Protocol\)](#).

**AutoDiscovery™** The technology in Visio® Enterprise used to find dynamically the devices attached to a network. The AutoDiscovery technology creates a database of network information, including device names, IP addresses, manufacturer, SNMP community strings used, and connectivity information. *See also* [IP \(Internet Protocol\)](#), [SNMP \(Simple Network Management Protocol\)](#).

**attribute** In a relational model, a column of a table. In an entity relationship model, a property of an entity. *See also* [entity](#), [entity relationship \(ER\) modeling](#), [relational model](#).

**bridge** Hardware that connects two LANs together. A bridge passes on messages between networks but does not perform any conversion of the network transmissions. *See also* [LAN \(local area network\)](#).

**cardinality** A characteristic of a relationship between two entities expressed as a ratio. The number of entity instances that participate in a relationship: one-to-zero-or-more (1:0), one-to-one (1:1), one-to-many (1:N), or many-to-many (N:M). *See also* [entity](#).

**catalog** A vendor-specific data dictionary of an existing relational database. *See also* [data dictionary](#), [relational database](#).

**category discriminator** An attribute in a parent (supertype) entity that is used to determine that category to which an instance of a child (subtype) entity belongs. *See also* [attribute](#), [generic entity \[IDEF1X\]](#).

**category entity [IDEF1X]** An entity whose instances represent a subset of a parent entity (generic entity). Also known as a subtype entity. *See also* [entity](#), [generic entity \[IDEF1X\]](#).

**child entity [IDEF1X]** The entity receiving a foreign key. A child entity receives a foreign key from a parent entity. *See also* [entity](#), [foreign key](#), [parent entity](#).

**circuit** A communication link between two or more points.

**composite data type** A type that combines one or more existing data types to create a new data type. Also called *complex type*.

**Data Definition Language** *See* [DDL \(Data Definition Language\)](#).

**data dictionary** A collection of information about the records, relationships, and other elements of a database.

**Data Manipulation Language** *See* [DML \(Data Manipulation Language\)](#).

**data migration** The use of data from an existing database to populate a new database schema. The final stage of the database reverse engineering process. *See also* [database schema](#), [reverse engineering](#).

**data model** A representation of the way data is stored in a database. The Visio Enterprise Database Model solution supports the object-relational and relational data models. *See also* [object-relational model](#), [relational model](#).

**database** Commonly, a collection of related tables.

**database driver** A software routine that enables two devices to communicate with each other. Visio Enterprise includes its own drivers to communicate with individual instances of databases.

**database management system** *See* [DBMS \(database management system\)](#).

**database re-engineering** A series of processes (reverse engineering, conceptual refinement, forward engineering, and data migration) resulting in the modification of a database schema, generally by first reconstructing the conceptual model that underlies it. *See also* [data migration](#), [forward engineering](#), [reverse engineering](#).

**database schema** The actual operational structure for a database application, built specifically for the data of that application.

**DBMS (database management system)** Application software for storing, manipulating, and retrieving data.

**DDL (Data Definition Language)** The language used to describe the organization of data in a database.

**dependent entity [IDEF1X]** An entity whose primary key includes one or more foreign key attributes. *See also* [attribute](#), [entity](#), [foreign key](#), [independent entity \[IDEF1X\]](#).

**dictionary** See [data dictionary](#).

**distinct type** A data type that is based on an existing data type with the same representation but distinct from the existing data type.

**DML (Data Manipulation Language)** The language used to retrieve, insert, delete, and update the data in a database.

**domain** A data type alias of a built-in, collection, or unnamed row type. It cannot be the name of a built-in, opaque, distinct, or named row type.

**driver** See [database driver](#).

**entity** In entity relationship modeling, anything about which information can be stored; for example, a person, concept, physical object, or event. Typically refers to a record structure. In a Visio Enterprise model, an entity can be represented visually by the Entity shape. See also [entity relationship \(ER\) modeling](#), [table](#).

**entity relationship (ER) modeling** A database design technique whose structural approach closely mirrors logical modeling, where attributes are grouped into entities and associations, which in turn are linked by relationships. See also [entity](#).

**external object type** In Visio Enterprise, a representation in one model diagram of an object type that is defined in another. A placeholder for an object type defined elsewhere. See also [model diagram](#).

**extracting** Creating a database model diagram from the schema of an existing database, a process performed by the Reverse Engineer Wizard in the Database Model solution. See also [database schema](#), [model diagram](#).

**field** A component of a row type. A field should not be confused with a column, which is associated with tables only.

**foreign key** A column or combination of columns in a child table whose entries must exist in the population of a primary key of the parent table. See also [primary key](#), [relational model](#).

**forward engineering** The process of implementing the model of a schema for a given database. An intermediate stage of database re-engineering. See also [database re-engineering](#), [database schema](#).

**frame relay** A wide area protocol used between network devices based on virtual circuits. A virtual circuit is a logical circuit created to communicate between two network devices.

**gateway** A computer used as a protocol conversion device. A gateway talks to each OSI stack's application layer when passing data or messages between two networks. See also [OSI \(Open System Interconnect\)](#).

**generation** A process that creates a DDL script from information in a database model diagram, and optionally passes it on to a database. *See also* [DDL \(Data Definition Language\)](#), [model diagram](#).

**generic entity [IDEF1X]** An entity whose instances are classified as one or more category entities. Also known as a supertype entity. *See also* [category entity \[IDEF1X\]](#), [entity](#).

**hub** A central connection point in a star-configured network.

**ICMP (Internet Control Message Protocol)** A protocol used to communicate IP status and error messages between hosts and routers. *See also* [IP \(Internet Protocol\)](#).

**IDEF1X notation** One of many entity relationship notations that represents data as entities with attributes that participate in relationships. A logical modeling notation supported by Visio Enterprise. *See also* [attribute](#), [entity relationship \(ER\) modeling](#), [Relational notation](#), [UML \(Unified Modeling Language\)](#).

**identifying relationship [IDEF1X]** A relationship in which the primary key attributes of the parent entity form part of the primary key of the child. *See also* [attribute](#), [nonidentifying relationship \[IDEF1X\]](#), [parent entity](#), [primary key](#).

**independent entity [IDEF1X]** An entity whose primary key contains no foreign key attributes. *See also* [attribute](#), [dependent entity \[IDEF1X\]](#), [foreign key](#), [primary key](#).

**interface** A connection between two network devices or hosts.

**IP (Internet Protocol)** A network layer protocol containing addressing and control information for network devices.

**LAN (local area network)** A LAN is composed of network devices tied together in a specific work area in the same general location.

**logical database model** A description of the structures within the application domain in terms of constructs supported by a relational database system. This description includes a series of relational tables and various constraints that apply both within and between these tables. *See also* [model diagram](#).

**MIB (management information base)** A database of managed objects accessed by network management protocols.

**model** A visual representation of data and its relationships.

**model diagram** A type of Visio Enterprise diagram that describes all or part of a database schema using the IDEF1X or Relational notation for viewing and editing. *See also* [database schema](#), [IDEF1X notation](#), [Relational notation](#).

**named row type** A group of typed fields that are defined under a single name. *See also* [field](#).

**nonidentifying relationship [IDEF1X]** A relationship in which the primary key attributes of the parent entity become non-key attributes of the child. *See also* [attribute](#), [identifying relationship \[IDEF1X\]](#), [parent entity](#), [primary key](#).

**normalization** A technique by which attributes are grouped into relations so as to avoid anomalies. *See also* [attribute](#), [relation](#).

**object-relational model** A conceptual modeling method that integrates many object-oriented and relational concepts. In Visio Enterprise, object-relational models support composite types, table inheritance, and type inheritance.

**Object Role Modeling (ORM)** A conceptual modeling method that views the application domain as a collection of objects playing roles. ORM is frequently contrasted with entity relationship modeling. *See also* [entity relationship \(ER\) modeling](#).

**Object Role Modeling diagram** A Visio Enterprise drawing type that describes all or part of a database schema as conceptual information using ORM diagrams (ORM documents) for viewing and editing. *See also* [database schema](#).

**optimization** The application of techniques to enhance performance.

**ORM** *See* [Object Role Modeling \(ORM\)](#).

**OSI (Open System Interconnect)** A communication standard that enables dissimilar networks to communicate. OSI is comprised of seven layers: Application, Presentation, Session, Transport, Network, Data Link, and Physical.

**parent entity** The entity contributing a foreign key. A parent entity contributes a foreign key to a child entity. *See also* [child entity \[IDEF1X\]](#), [entity](#), [foreign key](#).

**PING** A test of network connectivity to a remote device by sending an ICMP Echo to the device and then waiting for a reply. If it receives a reply, it can communicate with the device. *See also* [ICMP \(Internet Control Message Protocol\)](#).

**polling** The process where one device queries other devices for information. This process is used by the AutoDiscovery technology. *See also* [AutoDiscovery™](#).

**primary key** In a relational model, a column or column combination whose value can appear in no more than one row of the table at any given time. *See also* [relational model](#).

**probe** A device that is designed to help perform network management on a network segment. It monitors the local network and enables you to access information about the network.

**query** A request to a DBMS to produce specific information from data in a database. *See also* [DBMS \(database management system\)](#).

**re-engineering** *See* [database re-engineering](#).

**referential integrity** A relational database principle requiring foreign key values of the referencing (child) table either to be NULL or to match the primary key values of the referenced (parent) table. *See also* [foreign key](#), [primary key](#), [relational database](#).

**relation** A named table with a set of named columns that is populated by a set of unnamed rows.

**relational database** A database that appears as a collection of relations.

**relational model** A logical data model developed by E.F. Codd, in which all facts are stored in relations, and associations between relations are made by name rather than by predeclared access paths. The model allows data to be accessed a set at a time, rather than a record at a time, using relational operations such as natural join, selection, and union.

**Relational notation** A method of representing the components of a database as tables, columns, and relationships in a database model diagram. *See also* [IDEFIX notation](#).

**reverse engineering** The first stage of database re-engineering, where an existing database is used to reconstruct the data model that underlies it. *See also* [data model](#), [database re-engineering](#).

**router** A router connects two or more LAN segments together and can convert from different access methods (Token, Fiber, and so on), different message packet sizes, and different network speeds. When receiving a packet, the router determines how to forward that packet through the network. *See also* [LAN \(local area network\)](#).

**schema** *See* [database schema](#).

**SNMP (Simple Network Management Protocol)** A network management protocol used to manage IP network devices. SNMP uses community strings for authentication when managing devices. *See also* [IP \(Internet Protocol\)](#).

**SQL (Structured Query Language)** A commonly used Data Manipulation Language for databases. *See also* [DML \(Data Manipulation Language\)](#).

**stored procedure** A procedure created by analysts and stored in the data dictionary for repeated use by applications. *See also* [data dictionary](#).

**Structured Query Language** *See* [SQL \(Structured Query Language\)](#).

**subnet mask** A 32-bit address mask used in Internet Protocol to calculate a specific subnet. *See also* [IP \(Internet Protocol\)](#).

**switch** A network device that connects LANs and devices. *See also* [LAN \(local area network\)](#).

**synchronization** *See* [updating](#).

**table** Refers to data arranged in rows and columns. In relational database management systems, all information is stored in the form of tables. In a Visio Enterprise model, a table can be represented visually by the Entity shape. *See also* [entity](#).

**trigger** An event-driven stored procedure.

**UML (Unified Modeling Language)** A commonly used language that supports the creation of an integrated system model based on the Unified Modeling Language diagram types. Recently adopted as a standard by the Object Management Group, Inc.

**unnamed row type** A group of fields that are created using the ROW constructor. Unnamed row types cannot be assigned to tables.

**updating** 1. The process of changing a data model to match the database schema from which it was extracted, so that both model and database are in sync, a process performed by the Update Model Wizard in the Database Model solution. *See also* [data model](#), [database schema](#). 2. The process of changing a database to match a model diagram; for example, by adding and removing tables. The Update Database Wizard in the Database Model solution automatically updates a database. *See also* [model diagram](#).

**validation** The process that verifies the semantic correctness of a model. *See also* [model](#).

**view** In a database, a virtual table that can be queried. In a Visio Enterprise model, a view can be represented visually by the View shape. *See also* [table](#).

**WAN (wide area network)** A long distance network that usually connects multiple LANs. Computers connected to a WAN are often connected through dedicated high-speed phone lines (T1) or satellites. *See also* [LAN \(local area network\)](#).





# Index

## Symbols

- % (percent) sign 231
- %Author 231
- %Date 231
- %DocTitle 231
- %DocTitleLong 231
- %File 231

## Numerics

- 2-D shapes, connecting 52
- 3-way comparison. *See* Three-way comparison
- 32-bit ODBC driver 135, 136, 193

## A

- Abstractions. *See* Models
- Active Directory template 11
- ActiveQuery, installing 19–20
- Activity diagrams 95–96
- Address Resolution Protocol (ARP), defined 241
- Advanced Discovery Wizard 7, 26
- AK (notation) 168
- Alter, defined 241
- Altering a database. *See* Updating databases
- Alternate keys
  - generating 168
  - generating a unique index 168
- Appearance, customizing database model's 132, 168
- ARP (Address Resolution Protocol), defined 241
- Arrow (notation) 131, 165
- Associating
  - drivers 136
  - properties with UML elements 82–83
  - Visio Enterprise driver 194–195
- Attributes
  - category discriminator 150
  - defined 241
  - defining as alternate key 168

indexes 169

- reordering in an entity 153, 170
- reporting on 234, 236
- setting extended 171

### AutoDiscovery

- and devices 25
- database, managing 41–43
- defined 241

### AutoDiscovery And Layout solution

7, 9–10, 24

### Automatic code generation

- check clauses 185
- view definitions 163, 183

### Automatically discovering your network

23–44

## B

- Bachman template 18
- BASIC data types
  - accessing in UML 76
  - deleting in UML 76
- Basic Discovery Wizard 25, 26
- Basic Network template 11
- Basic, Visual. *See* Microsoft Visual Basic
- Booch OOD template 14, 61
- Borland. *See* Inprise
- Bridge, defined 241
- Browse Information file, generating 103
- Built-in data types. *See* also Physical data types
  - in Types window 122
  - reporting on 127, 238

## C

- C (notation) 166
- C++. *See* also Microsoft Visual C++
  - accessing in UML 76
  - data types, deleting in UML 76
  - reverse engineering 104
- Cardinality
  - assigning 164
  - defined 241

notation 132, 166, 168

Catalog, defined 241

### Categories

- locating equipment by 48
- working with 150, 150–151

Category discriminator 150, 151, 241

Category entity (IDEFIX), defined 242

Category shape 118, 150

### Category tables

- complete 150, 151
- incomplete 150, 151
- properties 151

Category To Child shape 118

### Changes

- detecting in a database 214, 216
- saving 34
- to network, updating 10

### Changing

- diagrams 33
- speed of interface 43

Chassis, adding to 50

### Check clauses

- adding to a column 154, 184, 186
- adding to a table 149, 184
- automatically generating code for 185
- reverse engineering 138
- saving in a mirror file 186

Check command (Database Model) 113, 124, 173, 202, 220

Checking database models. *See* Validating database models

Chen ERD template 14, 61

Child entity (IDEFIX), defined 242

CIR (Committed Information Rate), changing 43

Circuit, defined 242

Class diagrams. *See* Class static structure diagrams

Class static structure diagrams 93

Clauses, check. *See* Check clauses

Client/server database drivers 192

- Coad template. *See* Yourdon And Coad template
- Code
  - adding to a model 180–186
  - automatically generating for views 183
  - check clause, creating 184
  - defining view 182–183
  - description of 175
  - editing 176–178
  - for databases 175–189
  - function, creating 176, 181
  - global 177, 180, 181
  - local 177
  - local, check clauses 184
  - local, triggers 184
  - packages 176, 181
  - platform-specific 135, 176, 182
  - portable 176, 182
  - raw DDL 138, 176, 181
  - reverse engineering C++ source code 102–105
  - reverse engineering for databases 138
  - saving and synchronizing 186–187
  - stored procedures, creating 181
  - transferring to new target platform 188–189
  - trigger, creating 184
  - view definition, creating 161, 176, 182
- Code Editor 178–180
  - defining views in 182
  - setting preferences 179
- Code Selector. *See* Code window
- Code window 113, 122, 138, 183
  - opening 178
  - order code listed in 176
  - platform-specific code in 177, 188
  - viewing extracted code in 176
- Collaboration diagrams 92
- Collection types 159
- Columns
  - adding to a database model 152
  - adding to an index 170–171
  - attributes 172
  - check clause, creating 186
  - editing 153, 162
  - order in an index 171
  - removing 153, 170–171
  - reporting on 236
  - working with 152–154
- COM And OLE template 14, 61
- Combining lists with objects 30
- Command reference 4
- Committed Information Rate (CIR), changing 43

- Comparing tree view and diagrams 68
- Complete category 150
- Complete Compare. *See* Three-way comparison
- Complex. *See* Composite data types
- Component diagrams 97
- Components
  - installing database 16
  - modeling software 60
- Composite data types 118, 122, 154, 159, 242
  - reporting on 238
- Composite Type shape 118
- Conceptual
  - diagram, creating 86
  - model, creating with Object Role Modeling template 18
  - static structure diagrams 93
  - table report 236
  - view tools 7
- Configuring
  - 32-bit ODBC database drivers 193
  - Visio Enterprise driver 194–195
- Conforming to UML syntax 71
- Connecting
  - 2-D shapes 52
  - Relationship shapes 118, 134, 164, 165
  - Table Inheritance shapes 118
  - to a data source 136, 217
  - to a database 124
  - to Microsoft Repository 198–199
- Connection options 195
- Connector shapes
  - Category To Child 118
  - in relationships 118, 164, 165
  - Parent To Category 118
  - setting 151
- Control handles, screen tips 63
- Copying elements in tree view 80
- Corel Paradox 192
- Correcting errors and warnings 174
- Counter lists, device 31
- Creating
  - activity diagram 95–96
  - check clauses 184
  - class static structure diagram 93
  - collaboration diagram 92
  - component diagram 97
  - composite data type 160
  - conceptual diagram 86
  - data sources 196, 197
  - database indexes 168–171
  - database model 111, 116–119, 130–133, 133–135

- database report 227, 228
- databases 203
- deployment diagram 98
- logical network diagram 23
- multiple lists for same device 30
- network diagram 27–29
- ODBC data source 196–197
- package diagram 88–89
- physical network diagram 49–50
- reports 53–54, 111, 127–128
- RTF files, database report 234
- script to update database 220
- sequence diagram 90–91
- shortcut to shape 56
- statechart diagram 94
- statistical report 235
- table report 237
- trigger code 184
- types report 239
- use case diagram 85
- user-defined data types 158
- Customizing
  - database models 132, 142–143
  - labels 51
  - Microsoft Visual Basic 104
  - Microsoft Visual C++ 102
  - model's on-screen appearance 121
  - shapes 39
  - table notation 146

## D

- d (notation) 166
- Data
  - dictionary, defined 242
  - exporting to Microsoft Repository 101
  - importing from Microsoft Repository 100
  - migration, defined 242
  - model, importing 141–142
- Data Definition Language (DDL). *See* DDL (Data Definition Language)
- Data Link Connection Identifier (DLCI) lists 31
- Data Manipulation Language (DML) scripts 220, 243
- Data models
  - defined 242
  - IDEFIX notation 121, 132, 146, 166
  - importing 130, 141
  - Relational notation 121, 132, 146
- Data source
  - connecting to 136, 217
  - ODBC 196–197
- Data source name (DSN) 196

- Data types
  - adding in UML 77
  - BASIC, accessing 76
  - BASIC, contained in packages 66
  - BASIC, deleting in UML 76
  - built-in 122
  - C++, accessing 76
  - C++, contained in packages 66
  - C++, deleting 76
  - composite 118, 154, 159, 242
  - composite, defined 242
  - composite, reporting on 238
  - defining for database modeling 156
  - deleting in UML 76
  - listing in report 238–239
  - user-defined 156, 158
  - user-defined, reporting on 127, 238
  - working with 154
- Database Information Model (DBM)
  - 114, 115, 140, 193, 198
- Database management system (DBMS)
  - and database drivers 192–193
  - connecting to 191
  - defined 242
  - generate options 203
- Database model
  - collection data types 159
  - creating 116–119, 130–133, 133–135
  - customizing appearance 132
  - diagram, customizing 142–143
  - installing 16–17
  - logical, defined 244
  - process 110–115
  - refining 120–124
  - reverse engineering 119, 175–189
  - tables, adding 147
  - templates 110
  - validating 173–174
- Database schema
  - changes to 214
  - defined 242
  - errors when generating 206, 209
  - extracting 116, 135, 139
  - extracting from Microsoft Repository
    - 114, 135, 140, 217
  - forward engineering 202–204
  - generating 111, 125, 201, 202, 204, 207, 223
  - generating for multiple platforms 188
  - synchronizing 214, 218
  - updating 213, 216, 220, 223
  - validating 124, 173
- Database solutions, installing 16
- Databases
  - and portable code 176
  - client/server 217
  - client/server, connecting to 192
  - client/server, generating 202
  - components, installing 16
  - connecting to 124
  - creating 44, 203, 207
  - defined 242
  - deleting entries from 42
  - design tools 16
  - desktop, connecting to 192, 217
  - desktop, generating 202
  - detecting changes in 214
  - diagram solutions, installing 18
  - document options 143
  - drivers 136, 192–193, 242
  - forward engineering 202–204, 215, 220
  - generating 124–125, 202, 202–204, 207, 207–210, 223
  - generating for multiple platforms 188
  - information, translating 19
  - managing AutoDiscovery databases
    - 41–43
  - menu (Database) 113
  - Microsoft Repository,
    - connecting to 193
  - multiple AutoDiscovery databases
    - 43–44
  - re-engineering, defined 242
  - representing 130
  - reverse engineering 135–140
  - schema 111, 139, 140, 201, 223
  - scripts 220
  - solutions for modeling 16–20
  - specifying 44
  - synchronizing with models 126, 212, 214, 218
  - tables, listing in report 236–237
  - updating 125, 126, 211, 213, 215, 216, 220, 224
- DBM (Database Information Model)
  - 114, 115, 140, 193, 198
- DBMS. *See* Database management system
- DDL (Data Definition Language)
  - code 138, 181
  - defined 242
  - editing 204–207
  - generating 202–207
  - generating for models 203
  - stored procedures 181
  - tips for checking 220
- DDL scripts
  - adding code comments to 179
  - database code in 180
  - editing 207
  - generating 202–204, 216
  - generating a new database from
    - 203, 204
  - synchronizing 214
  - updating a database with 215, 220
- Default settings, overriding 133
- Defining
  - category in a model 151
  - data types 158
  - view code 182–183
- Deleting
  - C++ or BASIC data types in UML 76
  - columns 153, 170–171
  - custom shapes 39
  - elements 80, 81
  - entries from AutoDiscovery
    - database 42
  - relationships from a logical database
    - model 165
  - UML models 78
  - undeleting entries from the
    - AutoDiscovery database 42
- Dependent entity (IDEFIX), defined 243
- Dependent tables 146
- Deployment diagrams 98
- Design model (UML) 84, 91
- Design, User Interface solutions 63
- Desktop database drivers 192
- Detail, adding to diagrams 30–32
- Detecting changes in a database 214
- Developing Visio Solutions* 3
- Devices
  - and AutoDiscovery And Layout
    - solution 25
  - changing 33
  - changing text colors 40
  - discovering 23–44
  - lists 30, 31
  - renaming 42
  - shapes 36, 37
  - SysObjectID 35
- Diagnosing errors. *See* Errors
- Diagram solutions, installing 14–16, 18
- Diagrams
  - activity, creating 95–96
  - class static structure, creating 93
  - collaboration, creating 92
  - component, creating 97
  - conceptual, creating 86
  - creating 23, 27–29, 49–50, 85, 86, 88–89, 90–91, 92, 93, 94, 95–96, 97, 98
  - database model 129–143
  - database model, customizing
    - 142–143

- Diagrams (*continued*)
  - deployment, creating 98
  - detail, adding 30–32
  - elements, adding 80, 81
  - generation of logical network 9
  - hyperlinking network diagrams 29
  - list, adding 32
  - logical network, creating 23
  - network 9, 11, 12, 27, 44
  - package, creating 88–89
  - physical network, creating 49–50
  - sequence, creating 90–91
  - starting 10, 79
  - statechart, creating 94
  - tree view, comparing with 68
  - UML 62, 73–105
  - updating network 33–34
  - view of element, adding 81
  - working in 112
- Dialog box option help 63
- Dictionary, data defined 242
- Directory of manufacturer shapes 46
- Discovering and diagramming networks 23–44
- Discovery Wizard
  - Advanced 25, 26
  - Basic 25
- Discriminator, category. *See* Category discriminator
- Display options
  - choosing 122
  - setting for database model 132
- Displaying
  - collection types 160
  - equipment label 51
- Distinct type, defined 243
- DLCI (Data Link Connection Identifier)
  - lists 31
- DML (Data Manipulation Language)
  - scripts 220
- Document options, overriding 133
- Documentation window, UML
  - Navigator 65
- Documenting networks 9, 11, 12, 23, 44
- Domain model (UML) 84, 86
- Domain, defined 243
- Drawing files, UML 65
- Drawings, starting 10–12, 16–18
- Drivers
  - 32-bit ODBC 135, 193
  - 32-bit ODBC database,
    - configuring 193
  - and Database management system (DBMS) 192–193
  - associating 136
  - client/server database 192

- configuring 194–195
  - database 136, 176, 192, 193, 242
  - desktop database 192
  - enhanced database 192
  - extended attributes 169
  - Informix 192
  - Inprise database 192
  - installing database 193
  - Microsoft database 192
  - ODBC (Open Database Connectivity) 135, 192, 193
  - Sybase database 192
- Dropping relationships 165
- DSN (data source name) 196
- Duplicating elements 80, 81
- Duplicator, VNE Property 54–56

**E**

- Editing
  - code 176–178
  - column properties 153, 162
  - database indexes 168–171
  - database model 111, 120–124
  - DDL scripts 204–207
  - models 78
  - referential integrity settings 166–167
  - relationship properties 165
  - table properties 148, 151
- Editors
  - code 178–180
  - UML Property 74
- Elements 70, 79–83
- Endpoints, DLCI 31
- Engineering. *See* Forward engineering and Reverse engineering
- Enhanced database drivers 192
- Entities
  - attributes 134
  - category (IDEFIX), defined 242
  - child (IDEFIX), defined 242
  - creating 131, 133
  - defined 243
  - dependent (IDEFIX), defined 243
  - displaying 132, 134
  - drawing IDEFIX 112, 118, 131, 147
  - generic (IDEFIX), defined 244
  - independent (IDEFIX), defined 244
  - parent, defined 245
  - relationships 134, 164
- Entity relationship (ER) modeling,
  - defined 243
- Entity Relationship stencil 112, 116, 118, 119, 130, 131, 133, 134, 147, 150, 161

- Entries
  - deleting 42
  - inversion 168
- Equipment
  - installing Visio Network 11
  - labels 51
  - Locate Network 11, 47–48
  - shape, finding 48
- ER (entity relationship) modeling,
  - defined 243
- Errors
  - correcting in a database model 174, 206
  - correcting in a software model 71, 83
  - UML model 7
  - window, UML Navigator 65, 71
- ERwin files, importing 141
- Exporting
  - data to Microsoft Repository 101
  - reports 234
- Express-G template 15, 61
- Extended attributes
  - driver specific 169, 171, 172
  - setting for columns 154, 171
  - setting for indexes 169, 171
  - setting for tables 149, 171
  - setting for views 163, 171
- External object type, defined 243
- Extracting
  - code 139
  - data. *See* Reverse Engineering
  - defined 243
  - need to synchronize 126, 213
  - schema 135, 139, 140

**F**

- Field, defined 243
- Files
  - %File 231
  - Browse Information 103
  - ERwin 141
  - generating Browse Information 103
  - IMD, importing 141
  - importing 141
  - InfoModeler, importing 141
  - log 213
  - mirror 186–187
  - RTF, exporting reports to 234
  - UML drawing 65
- Finding equipment shape 48
- Fixing semantic errors 83
- FK (notation) 166
- Foreign key relationships
  - creating in a view 163

- displaying 132
- using Category shape 150
- Foreign keys
  - defined 243
  - detecting changes in 214
  - propagation options 142
  - referential integrity 166
  - reporting on 234, 236
  - setting column as 152
- Formatting database reports 230
- Forward engineering. *See also* Updating, databases 202–204
  - connecting to a data source 196, 197
- Frame relay
  - circuit, changing the CIR 43
  - defined 243
- Frequently Asked Questions (FAQs) 4
- Functions
  - editing 181
  - reverse engineering 138
- Fusion template 15, 61

## G

- Gane-Sarson DFD template 15, 61
- Gateway, defined 244
- Generate Wizard 202, 207–210
- Generating
  - Browse Information file 103
  - database reports 228–234
  - database schema 111, 201
  - databases 124–125, 202–204, 207–210
  - DDL scripts 204–207
  - defined 244
  - diagrams, logical network 9
- Generic database driver 193
- Generic entity (IDEFIX), defined 244
- Global code 177, 181
- Group header, formatting in database report 230
- Groups, formatting in database report 232
- Guide to ActiveQuery*, installing 20
- Guide to FORML*, installing 19
- Guide to VisioModeler*, installing 19
- Guided Update window 33

## H

- Headings, formatting in database report 230
- Help, online 4, 47, 63, 72
- Hiding an equipment label 51
- Hub, defined 244
- Hyperlinked diagram, creating 29

## I

- IBM database drivers 192
- ICMP (Internet Control Message Protocol), defined 244
- IDEFIX
  - attributes 131
  - category discriminators 150
  - display options 132, 143, 164, 168
  - drawing entities 112, 118, 131, 147
  - foreign key propagation options 143
  - generic entity, defined 244
  - referential integrity 121, 131, 164, 166
- IDEFIX notation
  - about 132, 146
  - changing 121
  - defined 244
  - for referential integrity 166
  - for tables 146
  - overriding display options 133
  - setting display options 132
- Identifying relationship 165, 244
- IE (notation) 168
- IMD files, importing 141
- Implementation model (UML) 84, 97
- Importing
  - data from Microsoft Repository 100
  - data model 141–142
  - database model 119
  - files 141
  - software models 100
- Incomplete category 150
- Independent entity (IDEFIX), defined 244
- Independent tables 146
- Indexes
  - attributes 149, 170
  - creating and editing 168–171
  - driver-specific extended
    - attributes 169
  - reporting on 234, 236
  - setting extended attributes 149
- InfoModeler files, importing 141
- Information model 99, 101
- Informix
  - DBMSs (database management systems) 203
  - drivers 192
- Inprise database drivers 192
- Installing
  - ActiveQuery 19–20
  - AutoDiscovery And Layout 9–10
  - database components 16
  - database diagram solutions 18
  - database drivers 193

- Database Modeling solution 16–17
- Diagram solutions 14–16, 18
- drivers 193
- Equipment, Visio Network 11
- network solutions 9, 11
- software diagram solutions 14–16
- solutions 9–10, 11
- templates 9–10, 11
- UML 13–14
- user manuals 19, 20
- Visio Enterprise 9
- Visio Network Equipment (VNE) 11
- VisioModeler 18–19
- Integrity, referential 166–168
- Interface
  - adding 33, 77
  - changing speed 43
  - defined 244
  - lists 32
- Internet Control Message Protocol (ICMP), defined 244
- Internet Protocol (IP), defined 244
- Interrupting a search 48
- Introduction to Visio Enterprise 5
- Inversion entries 152, 168
- IP (Internet Protocol), defined 244

## J

- Jackson template 15, 61
- Jacobson Use Cases template 15, 61
- Jet database, Microsoft 114, 115, 140, 198

## K

- Keys
  - alternate 168
  - foreign 7, 152, 166, 214, 243
  - primary 149, 152, 166, 168, 246

## L

- Labels, working with VNE 51
- LAN (local area network)
  - defined 244
  - links 25
- Language, Unified Modeling. *See* UML
- Layout solution, AutoDiscovery And 9–10
- Length of time needed for discovery 25
- Link speeds 25
- Lists
  - changes, saving 34
  - database tables in report 236–237
  - devices 30, 31
  - diagram detail 30–32
  - DLCIs 31

- Lists (*continued*)
  - interfaces 32
  - long versus short 30
- Local area network (LAN)
  - defined 244
  - links 25
- Local code 177
- Locate Network Equipment utility 11, 46, 47–48
- Locate window (VNE) 46, 47, 48
- Log files of database updates 213
- Logical database models
  - columns 152, 153
  - defined 244
  - errors, correcting 174
  - relationships, removing 165
  - warnings, correcting 174
- Logical network diagrams 9, 11, 23
- Logical validation of a database model 123
- Logical view tools 7
- LogicWorks. *See* PLATINUM ERwin files

## M

- Management information bases (MIBs) 9, 244
- Managing AutoDiscovery database 41–43
- Manual, installing user 19, 20
- Manufacturer shapes 46
- Many-to-many relationships 166
- Mapping network shapes 35
- Martin ERD template 15, 61
- Menus
  - Database 113
  - right-click 122
  - shortcut 24
- MIB. *See* Management information bases (MIBs)
- Microsoft 32-bit ODBC Data Source Administrator 135, 193
- Microsoft database drivers 192
- Microsoft Jet database 114, 115, 140, 198
- Microsoft Repository
  - connecting to 198–199
  - database models, storing and retrieving 114–115
  - extracting database schema 140
  - UML models, storing and retrieving 99–101
- Microsoft SQL code. *See* SQL (Structured Query Language) code
- Microsoft Visual Basic
  - customizing 104, 105
  - reverse engineering source code 102

- Microsoft Visual C++
  - customizing 102
  - data types, deleting in UML 76
  - reverse engineering source code 102, 104
- Visio UML Add-in 102–104
- Migration of data, defined 242
- Mirror file 186–187
- Model Diagram template, UML. *See* UML (Unified Modeling Language)
- Model diagrams
  - database, customizing 142–143
  - defined 245
- Model, Update Wizard 218–219
- Modeling
  - databases 16–20, 110–115, 129–143
  - in Visio Enterprise 6–8
  - networks 23–44
  - new system with UML 76
  - preferences, setting 143
  - software components 60
  - software, solutions 13, 59–72
  - systems using UML 74–83
- Modeling approaches
  - IDEFIX 121
  - object-relational 118, 130–131
  - Relational 112, 118, 130, 131
- Models
  - adding UML 78
  - and systems 66
  - appearance, customizing for databases 121
  - category, defining for databases 151
  - code, adding for databases 180–186
  - conceptual, creating with Object Role Modeling template 18
  - data, defined 242
  - data, importing for databases 141–142
  - data, importing for UML 100
  - database, creating 116–119, 130–133, 133–135
  - DDL script for databases, generating 203
  - defined 244
  - deleting in UML 78
  - editing in UML 78
  - elements, adding UML 79
  - sharing with other Visio users 8
  - starting new UML 14, 79
  - storing and retrieving for UML 99–101
  - synchronizing databases 212–218

- updating for databases 125–126, 211, 214
- validating for databases 123, 173–174
- viewing 67
- Modifying network shapes 54–56
- Modules, adding to a chassis 50
- Moving
  - columns 153
  - elements in tree view 80
- Multiple AutoDiscovery databases 43–44

## N

- N (notation) 166
- Named row type, defined 245
- Nassi-Schneiderman template 15, 61
- Natural language, translating database information into 19
- Navigator, UML 65, 71, 74
- NDS (Novell Directory Services) Solution 12
- Network
  - changing 10, 33
  - devices, changing text colors 40
  - diagrams 9, 11, 12, 24, 27–29, 49–50
  - discovering 23–44
  - equipment. *See* Visio Network Equipment (VNE) and Locate Network Equipment utility
  - shapes 38, 46–47, 54–56
  - solutions for diagramming and documenting 9, 11, 12
- Network diagrams, logical
  - adding detail to 30–32
  - changing text colors 40
  - creating 27–29
  - discovering devices 24
  - hyperlinking 29
  - updating 33–34
  - using other shapes 36
  - using VNE shapes 35
- New Report Wizard. *See* Report Wizard
- Nonidentifying relationship 165, 245
- Normalization, defined 245
- Notation
  - cardinality 166
  - customizing table 146
  - description 63
  - IDEFIX 132, 146, 244
  - referential integrity 166
  - Relational 132, 246
  - table 146
- Novell Directory Services (NDS) Solution 12
- NULL values 167



## O

Object Management Group (OMG)  
13, 63

Object Relational stencil 112, 116,  
118, 119, 130, 131, 133, 134, 147,  
150, 161

Object Role Modeling (ORM)  
defined 245  
diagram 245  
template 18

Object-relational databases, representing  
130

Object-relational model, defined 245

ODBC (Open Database Connectivity)  
data source 196–197  
drivers 135, 192, 193

Office Layout For Networks template 11

OLE template. *See* COM And OLE  
template

OMG *Unified Modeling Language  
Specification* 64

OMG. *See* Object Management Group

On-screen appearance, customizing 121

One-page diagram, creating 27

Online help 4, 47, 63, 72

Open Database Connectivity (ODBC)  
data source 196–197  
drivers 135, 192, 193

Open System Interconnect (OSI),  
defined 245

Opening  
AutoDiscovery And Layout  
template 24  
database model solution 130  
UML solution 65

Optimization, defined 245

Options  
choosing database display 122  
connection 195  
database document 143  
DBMS generate 203  
display for database model 132  
network installation 11  
overriding 133

Oracle DBMSs 192, 203

ORM. *See* Object Role Modeling (ORM)

OSI (Open System Interconnect),  
defined 245

Output window  
code transferring messages in 219  
database updating messages in 188,  
222, 224  
error messages in 124, 173, 174,  
206, 209  
importing messages in 142

opening 113

reverse engineering messages in  
140, 141

Overriding document options 133

## P

P (notation) 168

Package diagrams 88–89

Packages (Oracle code type) 176, 181

Packages in UML, using 69

Pages, formatting in database report  
230, 231, 232

Paradox. *See* Corel Paradox

Parent entity, defined 245

Parent To Category shape 118

PDF format of user manuals, installing  
19, 20

Percent (%) sign 231

Physical  
database data types 155  
database schema. *See* Database  
schema  
database table report 236  
network diagrams, creating 49–50  
validation of a database model 123  
view tools 7

PING  
and devices 25, 26  
defined 245

PK (notation) 168

Platform-specific code 176, 188–189

PLATINUM ERwin files, importing 141

Polling, defined 245

Portable code 155, 156, 176

Predefined shapes 74

Preferences, setting 143

Prefixes in database names 143

Previewing database reports 233

Primary keys  
defined 246  
generating an index 149, 152, 168  
propagating changes 132, 167  
referential integrity 166  
setting column as 149, 152–153  
synchronizing changes to 214

Probe, defined 246

Procedures, stored 181

Program Structure templates 15, 61

Programming language reference 4

Programs, Setup 11

Properties  
adding 55, 70  
and UML elements 70  
associating with UML elements  
82–83

column, editing 153, 162

editing 148, 151, 165, 170

of a shape 53

relationship 165

Property Duplicator, VNE 54–56

Property Editors, UML 74

Pushpin 149

## Q

Query, defined 246

## R

R (notation) 166

Rack, adding a chassis to 50

Raw DDL code 176, 181

Re-engineering a database  
defined 242  
forward engineering 215, 220  
reverse engineering 116

Re-engineering tools 16

Real-Time Object-Oriented Modeling  
(ROOM) template 15, 62

Records, report 232

References, online 4

Referential integrity  
defined 246  
displaying 132, 164, 166  
notation 166  
setting 166–168

Refining database model 111, 120–124

Relation, defined 246

Relational databases  
defined 246  
representing 130

Relational notation  
defined 246  
display option 132  
displaying tables in 146, 164

Relationship shape 118

Relationships  
adding 164  
attributes 172  
cardinality 131, 165, 166  
defining a foreign key 7  
properties, editing 165  
removing 165  
restoring 214  
type inheritance 118

Relay. *See* Frame relay

Remote devices and AutoDiscovery And  
Layout solution 25

Removing  
columns 153, 170–171  
custom label 51  
index 169

- Removing (*continued*)
  - relationships 165
  - table from database model 147
  - user-defined data types 158
- Renaming a device 42
- Reordering
  - columns 153
  - views in Code window 183
- Repairing foreign key relationships 214
- Report window 128, 236, 237, 238, 239
- Report Wizard 229
- Reports
  - creating 53–54, 111, 127–128, 227
  - data types 238–239
  - generating for database 228–234
  - listing tables in 236–237
  - previewing 233
  - statistics, summarizing 234–235
  - using conceptual names 236
  - using database physical names 236
- Repositioning labels 51
- Repository Type Information Model 99, 101
- Repository. *See* Microsoft Repository and SQL Server repository database
- Representing databases 130
- Restoring foreign key relationships 214
- Retrieving UML models 99–101
- Reverse engineering
  - connecting to a data source 136, 139, 196–197, 217
  - database model 119
  - databases 135, 138, 140, 175, 189
  - defined 246
  - from Microsoft Repository 140, 217
  - source code 102–105
- Reverse Engineering Wizard 139
- Rich Text Format (RTF) files, creating reports 234
- Right-click menus 122
- ROOM template 15, 62
- Router, defined 246
- Row type
  - named, defined 245
  - unnamed, defined 247
- Rumbaugh OMT template 15, 62

## S

- Saving
  - code 186–187
  - modified shape on a stencil 56
- Scale of network shapes 46
- Schema
  - changes to 214
  - database, defined 242
  - errors when generating 206, 209

- extracting 139, 140
- forward engineering 202–204
- generating database 111, 125, 201, 202, 204, 207, 223
- generating for multiple platforms 188
- synchronizing 214, 218
- updating database 213, 216, 220, 223
- validating 123, 173
- Screen tips for toolbar buttons 63
- Script
  - generating DDL 202–207
  - to update database 220
- Searches, interrupting 48
- Semantic errors
  - checking 74, 83
  - UML 71, 74, 83
- Semantics 63
- Sequence diagrams 90–91
- Setting
  - data types 155
  - display options for database model 132
  - extended attributes 171–172
  - referential integrity 166–168
- Setup program 11
- Shapes
  - adding custom labels 51
  - and templates 8
  - connecting 2-D 52
  - custom 39
  - device 36, 37
  - finding equipment 48
  - help, getting 4, 63
  - mapping 35
  - modeling a system 74
  - modifying network 54–56
  - network 38, 46–47
  - Network Equipment Directory 46
  - notation 63
  - predefined 74
  - properties 53
  - returning to default shapes 39
  - setting category connector 151
  - VNE 11, 35–39
- Sharing models with other Visio users 8
- Shlaer-Mellor OOA template 15, 62
- Shortcut
  - creating for shapes 56
  - creating for templates 76
  - creating keyboard 103
  - menus 24, 122
- Simple data types 154
- Simple Network Management Protocol (SNMP). *See* SNMP (Simple Network Management Protocol)

- SNMP (Simple Network Management Protocol) 9, 25, 26, 246
- Software
  - components, modeling 60
  - diagram solutions, installing 14–16
  - modeling solutions 13, 59–72
- Solutions
  - Active Directory 11
  - and shapes 8
  - AutoDiscovery And Layout 9–10, 24
  - Bachman 18
  - Basic Network 11
  - Booch OOD 14, 61
  - Chen ERD 14, 61
  - COM And OLE 14, 61
  - Database Diagram, installing 18
  - Database Modeling 110
  - documenting networks 9, 11, 12
  - Express-G 15, 61
  - Fusion 15, 61
  - Gane-Sarson DFD 15, 61
  - help, getting 4, 63
  - in Visio Enterprise 7
  - installing 9–10, 11
  - installing diagram 11, 14–16
  - Jackson 15, 61
  - Jacobson Use Cases 15, 61
  - Logical Network Diagram 11
  - Martin ERD 15, 61
  - modeling databases 16–20
  - Nassi-Schneiderman 15, 61
  - NDS (Novell Directory Services) 12
  - Network Diagram 12
  - Novell Directory Services (NDS) 12
  - Object Role Modeling 18
  - Office Layout For Networks 11
  - Program Structure 15, 61
  - Real-Time Object-Oriented Modeling (ROOM) 15, 62
  - ROOM 15, 62
  - Rumbaugh OMT 15, 62
  - Shlaer-Mellor OOA 15, 62
  - shortcuts for creating templates 76
  - software modeling 13, 61–63
  - SSADM 15, 62
  - System Structure 15, 62
  - UML Model Diagram 7, 62, 74–83
  - User Interface Design 63
  - Visio Enterprise 7
  - Visio Network Equipment (VNE) 11
  - VNE 11, 46
  - Windows User Interface 15, 62
  - Yourdon And Coad 15, 62
- Sorting database reports 230, 232, 238



- Source
  - code, reverse engineering for
    - databases 175–189
  - code, reverse engineering for UML 102–105
  - connecting to data 217
  - data. *See* Data source
  - ODBC data 196–197
- Speed of interface, changing 43
- SQL (Structured Query Language) code
  - as portable code 176
  - defined 247
  - editing 179
  - reverse engineering 138
  - view, editing 161, 163
- SQL Server repository database 198
- SSADM template 15, 62
- Starting
  - diagram 79
  - drawing 10–12, 16–18
- Statechart diagrams 94
- Static structure diagrams 93
- Statistics, summarizing in report 234–235
- Stencils 8, 11, 39, 56
- Stored procedures 181
  - defined 247
  - reverse engineering 138
  - saving in a mirror file 186
- Storing UML models 99–101
- Structured Query Language (SQL).
  - See* SQL (Structured Query Language) code
- Subnet mask, defined 247
- Summarizing statistics in report 234–235
- Switch, defined 247
- Sybase database drivers 192
- Synchronizing
  - model with database 212
- Synchronizing. *See also* Updating
  - code 186–187
  - database with model 218
  - model and database 212–218
- Syntax, UML 71
- SysObjectID 35
- System Structure template 15, 62
- Systems
  - and models 66
  - and UML drawing files 65
  - modeling with UML 74–83

## T

- Table Inheritance shape 118
- Table Selector. *See* Tables & Views window

- Tables
  - independent 146
- Tables & Views window 113, 122, 136, 140
  - removing objects from 147, 161
  - views in 182
- Tables. *See also* Entities
  - attributes 172
  - check clause, creating 185
  - defined 247
  - display options 121, 132, 146
  - forward engineering 204, 215
  - listing in report 236–237
  - reporting on 236
  - reverse engineering 139
  - synchronizing changes in 126, 213
  - using the Entity shape 131, 134, 147
  - working with 146–151
- Target platform, transferring code to 188–189
- Templates
  - Active Directory 11
  - and shapes 8
  - AutoDiscovery And Layout 9–10, 24
  - Bachman 18
  - Basic Network 11
  - Booch OOD 14, 61
  - Chen ERD 14, 61
  - COM And OLE 14, 61
  - Database Diagram, installing 18
  - Database Modeling 110
  - documenting networks 9, 11, 12
  - Express-G 15, 61
  - Fusion 15, 61
  - Gane-Sarson DFD 15, 61
  - help, getting 4, 63
  - in Visio Enterprise 7
  - installing 9–10, 11
  - installing diagram 11
  - installing diagram solutions 14–16
  - Jackson 15, 61
  - Jacobson Use Cases 15, 61
  - Logical Network Diagram 11
  - Martin ERD 15, 61
  - modeling databases 16–20
  - Nassi-Schneiderman 15, 61
  - NDS (Novell Directory Services) 12
  - Network Diagram 12
  - Novell Directory Services (NDS) 12
  - Object Role Modeling 18
  - Office Layout For Networks 11
  - Program Structure 15, 61
  - Real-Time Object-Oriented Modeling (ROOM) 15, 62
  - ROOM 15, 62

- Rumbaugh OMT 15, 62
- Shlaer-Mellor OOA 15, 62
- shortcuts for creating 76
- software modeling 13, 61–63
- SSADM 15, 62
- System Structure 15, 62
- UML Model Diagram 7, 62, 74–83
- User Interface Design 63
- Visio Enterprise 7
- Visio Network Equipment (VNE) 11
- VNE 11, 46
- Windows User Interface 15, 62
- Yourdon And Coad 15, 62
- Three-way comparison 212
- Time required for discovery of network 25
- Tips
  - adding elements 80
  - connection options 195
  - creating keyboard shortcuts 103
  - data types, controlling 76
  - database model diagram,
    - synchronizing 140
  - DDL scripts, checking 220
  - deleting elements 80, 81
  - deleting models 78
  - diagrams, creating new 79
  - diagrams, reading 31
  - displaying for toolbar buttons 4
  - displaying Relationship
    - Properties 167
  - duplicating elements 80, 81
  - editing model properties 78
  - errors, correcting 206
  - exporting data 101
  - importing models 100
  - interfaces, adding 77
  - Locate Network Equipment utility,
    - opening 48
  - network shapes 46–47
  - overriding document options 133
  - properties, adding to modified
    - shape 55
  - Pushpin 149
  - shortcut to UML Model Diagram 76
  - table properties, displaying 148
  - Toolbar and Tools menu 104
  - toolbar buttons and control
    - handles 63
  - user-defined data types, deleting for
    - databases 158
  - View Properties, displaying 163
- Titles, formatting in database report 230
- Tool reference 4

- Toolbar
  - and Tools menu 104
  - AutoDiscovery And Layout 24
  - buttons 63
  - displaying tips for buttons 4
- Tools
  - database design 16
  - in Visio Enterprise 7
  - modeling a system 74
  - re-engineering 16
- Transferring code to new target platform 188–189
- Translating database information 19
- Tree view 65, 68, 80
- Trigger code
  - creating 177, 184
  - in tables 149
  - reverse engineering 138
  - saving in a mirror file 186
- Trigger, defined 247
- Type Inheritance shape 118
- Type Selector. *See* Types window
- Types
  - collection and composite 159
  - data. *See* Data types
  - distinct, defined 243
  - report 238
  - window 122

**U**

- u (notation) 166
- UML (Unified Modeling Language)
  - adding elements to model 79
  - associating property values with elements 70, 82–83
  - defined 247
  - deleting from a diagram 81
  - diagram types 67, 75, 79
  - drawing files 65
  - duplicating 80, 81
  - element properties 70
  - help, online 72
  - Model Diagram template 7, 62
  - modeling system 74–83
  - models, conceptual overview 63–64, 75
  - models, creating 76–81
  - models, errors in 7
  - Navigator 65, 74
  - Property Editors 70, 74, 82
  - shortcut to UML Model Diagram 76
  - starting a diagram 14
  - syntax 71
- UML Navigator 65, 71
- Undeleting entries from the AutoDiscovery database 42

- Unique indexes 152, 168
- Unnamed row type, defined 247
- Update Database Wizard 220–224
- Update Model Wizard 218–219
- Update window, Guided 33
- Updating. *See also* Synchronizing databases 125–126, 211, 212, 215, 220
  - defined 247
  - models 111, 125–126, 211, 214
  - network changes 10
  - network diagrams 33–34
  - schema, database 223
- URL for Visio home page 4
- Use case diagrams 85
- User interface design solutions 63
- User manuals, installing 19, 20
- User-defined data types 158
- Using Visio Enterprise* 3
- Utility, Locate Network Equipment 11, 47–48

## V

- Validating database models 123, 173–174
- Validation, defined 247
- Verbalizer window 149
- View code, defining 182–183
- View definitions
  - adding columns to 162
  - automatically generating code for 183
  - based on other views 182, 183
  - creating 161
  - editing properties of 163
  - setting extended attributes for 172
- View shape 118, 161, 182
- Viewing
  - code 176–178
  - models 67
  - properties of a shape 53
- Views. *See also* View definitions
  - adding to UML diagram 81
  - customizing appearance of 121
  - defined 247
  - reordering in Code window 183
  - reverse engineering 138
  - working with 161–163
- Visio Enterprise
  - diagram 112
  - features 8
  - help, getting 4, 63
  - installing 9
  - solutions 7
  - Web site 4

- Visio Knowledge Base, locating 4
- Visio Network Equipment. *See also* Locate Network Equipment utility
  - description 7, 45
  - directory 46, 48
  - installing 11
  - shapes 35–39, 46–47
  - templates 11
- Visio UML Add-in 102–104
- VisioModeler, installing 18–19
- Visual Basic. *See* Microsoft Visual Basic
- Visual Fox Pro 3.0 203
- VNE Directory 48
- VNE Property Duplicator 54–56
- VNE. *See* Visio Network Equipment (VNE)

## W

- WAN (wide area network)
  - defined 247
  - links 25
- Warnings, correcting 174, 206
- Web site, Visio 4
- Wide area network (WAN)
  - defined 247
  - links 25
- Windows
  - Code 122
  - code 177–178, 183
  - Documentation 65
  - Errors (UML) 65, 71
  - Guided Update 33
  - Locate (VNE) 46, 47, 48
  - Output. *See* Output window
  - Report 128, 236, 237, 238, 239
  - report preview 233
  - Tables & Views 122
  - Types 122
  - User Interface template 15, 62

- Wizards
  - Advanced Discovery 26
  - Basic Discovery 25, 26
  - Generate 202, 207–210
  - Report 229
  - Reverse Engineer 139
  - Update Database 176, 181, 220–224
  - Update Model 218–219
- Working with labels 51

## Y

- Yourdon And Coad template 15, 62

## Z

- Z (notation) 168