



**Working with the
System Architect V10 Repository**

A Telelogic Working Paper

Working with the System Architect V10 Repository is a Telelogic whitepaper. Document version 1.2, published July, 2005. This whitepaper is intended to be used with System Architect V9.1, V10, or later.

Table of Contents

Table of Contents	3
Introduction	5
Encyclopedia Location and Structure	5
SQL Server 2000.....	5
MSDE	5
Location of the Encyclopedia	5
Location of the Server	6
Location of the Database (or Encyclopedia)	7
Location of the UDL File.....	8
Structure of the Encyclopedia.....	8
Going Deeper – The Encyclopedia’s Internal Tables	10
Creating or Opening an Encyclopedia	19
Creating an Encyclopedia.....	19
Opening an Encyclopedia.....	20
Opening Prebuilt, Sample Encyclopedias Shipped With SA	21
Moving or Making Copies of Encyclopedias	23
Option 1 (Preferable): Making and Sending a Backup (.BAK) File.....	23
Option 2: Unattaching and Sending the .MDF File.....	24
Customizing an Encyclopedia	24
Modifying the Encyclopedia’s Metamodel.....	25
Specifying Your Own Images for Symbols on Diagrams	26
Specifying Your Own Images for Existing Encyclopedias	26
Specifying Your Own Images for New Encyclopedias	28
Copying and Pasting Symbols Onto Diagrams	29
Picture File Copy.....	29
Picture File Reference.....	29
Maintaining An Encyclopedia	30
Appendix A: Attaching and Opening an Encyclopedia	31
In this section, we provide detailed steps for attaching and opening encyclopedias.	31
Overall Steps	31
Attaching an Encyclopedia to Your Server.....	31
Create a UDL File for the Encyclopedia.....	32
Appendix B: ENTITY Table Contents.....	35
Appendix C: Relationship Numbers	37

Introduction

System Architect V9 provides a new underlying storage mechanism for its repository, based on either SQL Server 2000 Standard Edition or Microsoft Server Desktop Engine 2000 (MSDE 2000). This is different than previous versions of System Architect, which used a proprietary, flat file repository structure.

This paper introduces you to the new System Architect repository. It explains where the repository is stored, how you access it, how to work with files (USRPROPS.TXT and SAPROPS.CFG) used to customize a repository's metamodel in the new environment, and how to maintain a SQL Server encyclopedia. You are provided with details on the structure of the repository, and tips on performing such tasks as sending a copy of an encyclopedia to someone.

Encyclopedia Location and Structure

In System Architect, you create and store all of your models, including diagrams and definitions, in a repository. This repository is also called an encyclopedia, and each encyclopedia is a database in SQL Server 2000 or MSDE 2000.

SQL Server 2000

You may create an encyclopedia on SQL Server 2000 in a network environment, and utilize SQL Server's capabilities for access control, rollback, and database management. You can only create an encyclopedia on SQL Server 2000 in a network environment if you have sufficient access rights, as granted by your system administrator.

MSDE

Instead of SQL Server 2000, you may alternately use Microsoft Server Desktop Engine 2000 (MSDE 2000) for network encyclopedias. MSDE does not have the scalability of SQL Server 2000.

Most often, you use MSDE 2000 to create and work in encyclopedias for single-user use on your desktop or laptop. Installation of System Architect includes installation of a local instance of MSDE.

Location of the Encyclopedia

When you create an encyclopedia with System Architect V9, you specify three things:

1. **The server** that the encyclopedia will be created in (either one on the network or an MSDE instance on your local machine),
2. A **name for the database**, and
3. A **name for a Universal Data Locator (UDL) file** that will point at the database.

As mentioned in the previous section, the server will either be a remote SQL Server 2000 or a local server running MSDE 2000.

System Architect Encyclopedia

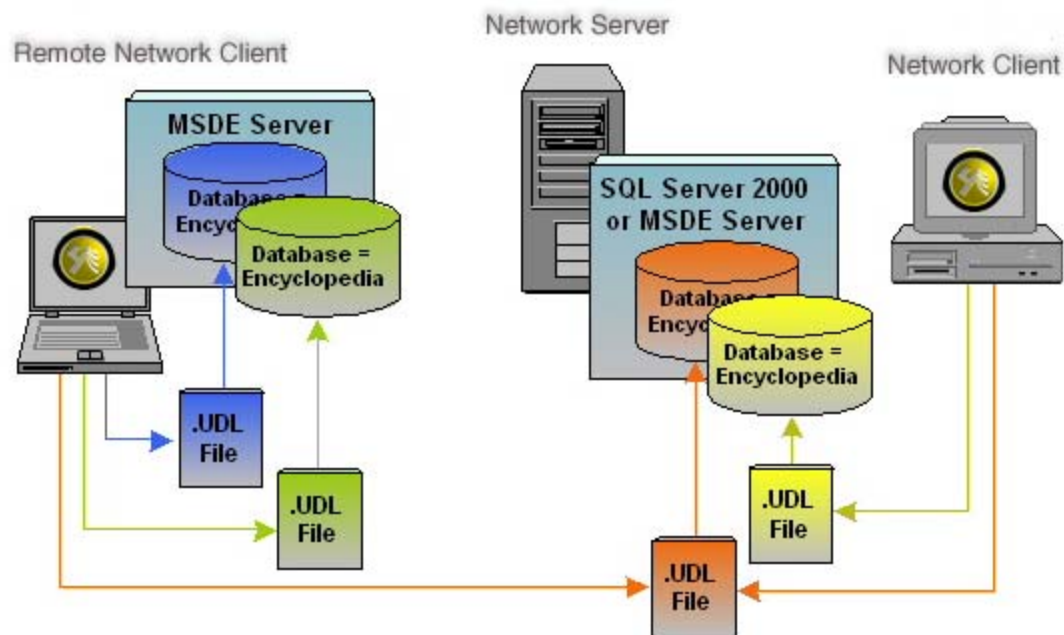


Figure 1. Possible Locations of System Architect Encyclopedias

Location of the Server

The location of this server is transparent to you. The System Architect Encyclopedia Open dialog will present you with a list of all applicable servers that it sees across the network. This includes your server, the network server, and any local MSDE servers located on other user's machines.

To connect to a server and use it, you need the appropriate access rights. By default, you have sufficient access to your local MSDE instance if you are a member of the local administrator's group on your PC. See the document titled **System Architect V9 Repository Access and Security**, available on http://www.popkin.com/customers/customer_service_center/downloads/whitepaper/index.htm, for details about setting up various level of Access in SQL Server 2000. If you need to make changes an MSDE 2000 instance, you may use the same techniques described in that document. You may also consult the following "Tip of the Week" article: [Granting a User Access and Assigning a Role On MSDE Using the Osql Utility](http://www.popkin.com/customers/customer_service_center/tip_of_the_week/03_03_03/tipofweek.htm) (http://www.popkin.com/customers/customer_service_center/tip_of_the_week/03_03_03/tipofweek.htm).

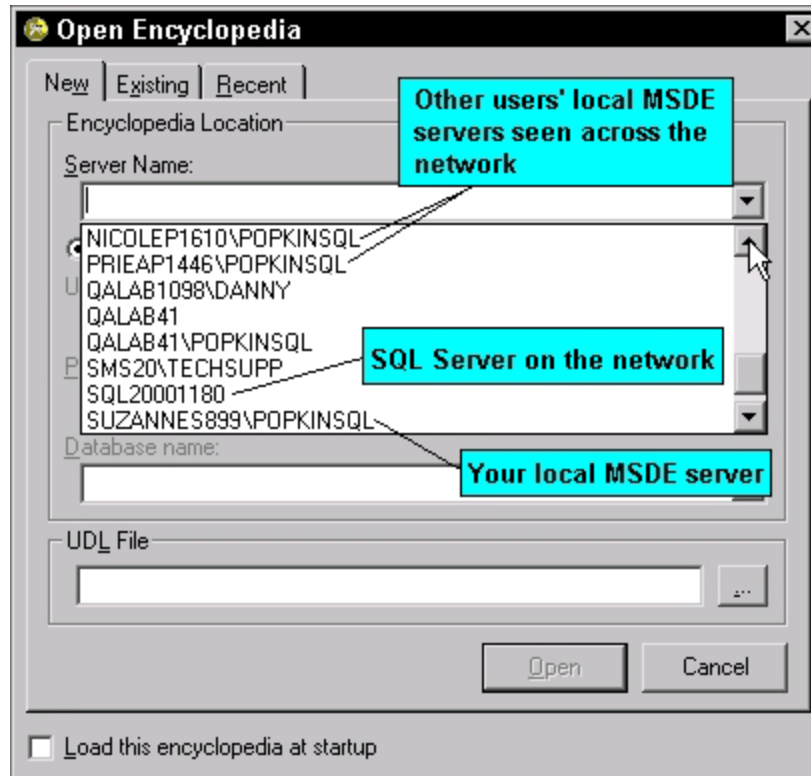


Figure 2. Servers you see in Encyclopedia Open dialog.

TIP



Although you can see other user's local MSDE servers, you will probably not have permission to access them unless your network administrator has specifically given you rights. You will know whether or not you have access rights to a server when, after selecting it, you try to select a database on that server in the Database Name field. If you don't have access, you will get a login failure error. If you do have access, you will get a list of all databases (encyclopedias) on that server.

Location of the Database (or Encyclopedia)

Similarly, the location of the database (or encyclopedia) is transparent to you. You simply need to specify its name in the System Architect Encyclopedia Open dialog. The database (or encyclopedia) itself is actually comprised of an MDF file with some associated files, stored in a directory on your computer or the network. These files are typically stored in the Data directory of SQL Server's MSSQL\$POPKINSQL directory, as shown in Figure 2.

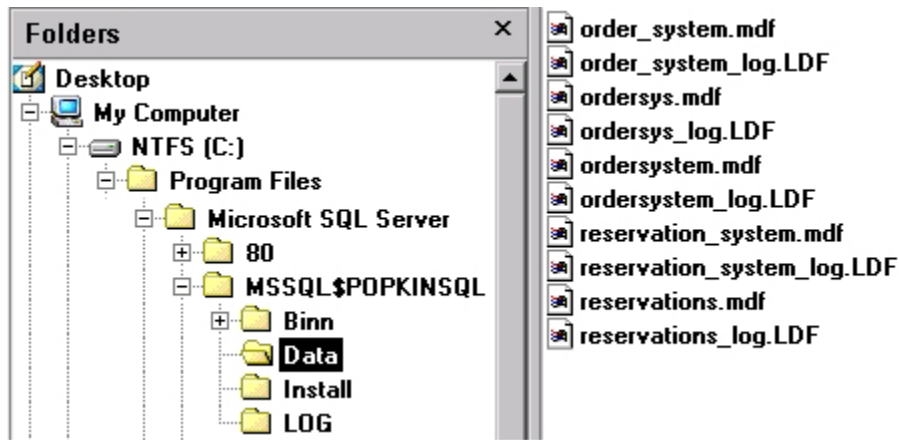


Figure 2. Default Location of System Architect Encyclopedia MDF File

TIP MSDE's default path to hold MDF files is in the Data subdirectory of the <C>:\Program Files\Microsoft SQL Server\MSSQL\$POPKINSQL.

Location of the UDL File

The location of the UDL file is important to you, since it is a pointer to the encyclopedia = database. When opening an encyclopedia, you simply open the UDL file through System Architect's **Open Encyclopedia** dialog – you do not specify the MDF file, the UDL file does this for you.

You may specify that the UDL file gets created in any directory. The UDL file is what you will use to access your encyclopedia in the future, so it is best to store it in a convenient location. You can store the UDL file in your **My Documents** directory, or any other directory in which you decide to store your encyclopedia UDL files.

TIP The UDL (or Universal Data Locator) file is simply a pointer to the database on a server. It can be located anywhere (i.e., MyDocuments). It is a good idea to place UDL files in a location that you can easily access and remember. It is also recommended that you name the UDL file that same as your encyclopedia database name so you can easily identify it. .

Structure of the Encyclopedia

As we have stated, the encyclopedia is a database in SQL Server 2000 or MSDE. Each database has a corresponding .UDL file that you store in a directory that you choose. Opening the .UDL file opens the actual database (or encyclopedia).

Encyclopedia Is an MDF File

Underneath the covers, the database (or encyclopedia) itself is physically an MDF file. This MDF file is stored, by default, in SQL Server's Data directory. If you are

running a local instance of MSDE, this directory is typically in your <C>:\Program Files\Microsoft SQL Server\MSSQL\Data directory. For every MDF file stored in this directory, there is also a corresponding _log.LDF file. The .LDF file contains a log of transactions that occur on its affiliated database (MDF file). The file structure that composes a System Architect encyclopedia is shown in Figure 3 below.

WARNING An MDF file is an SQL Server database file. It has hooks to the SQL Server service. Because of this, you cannot update, delete, copy or move this file. For instructions on how to move or make copies of encyclopedias, please see the section labeled appropriately later in this paper.

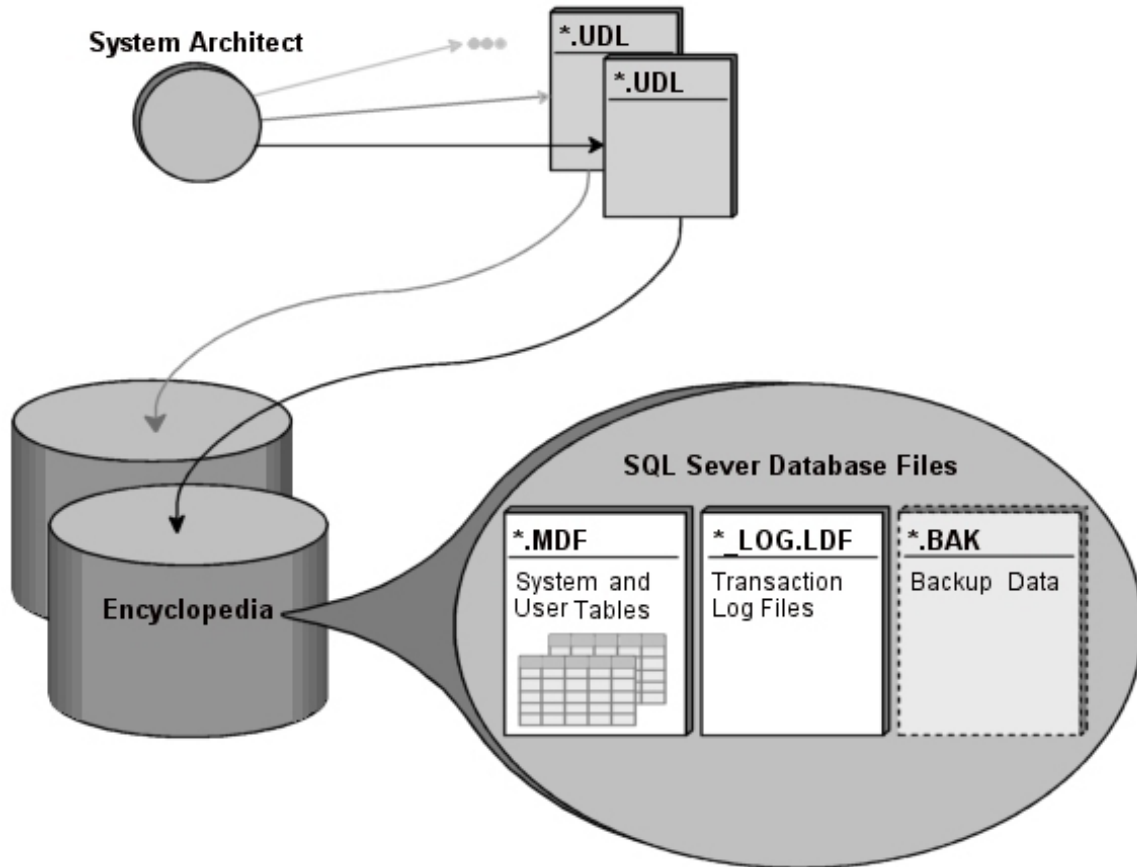



Figure 3. Structure of System Architect Encyclopedia

Going Deeper – The Encyclopedia’s Internal Tables

First let’s just make it clear that you don’t need to read anything more in this section. The actual makeup of an encyclopedia is advanced material that you don’t need to know. It is only useful for advanced users or the curious.

An encyclopedia is a SQL Server database that exists physically as an MDF file. Within that database there are tables. The tables that comprise a System Architect encyclopedia are as follows:

- ENTITY table
- RELATIONSHIP table
- FILES table
- PROPERTIES table
- IDGENERATOR table
- ERROR_LOG table
- INFO_LOG table
- NETWORK_LOG table

WARNING



An encyclopedia’s schema is designed to optimize the performance of System Architect and not for the typical end user to interact with directly, independent of the tool’s reporting interfaces. In fact, any modification of an encyclopedia database’s schema will render the database unusable by System Architect.

The tables that store the encyclopedia data are not fully normalized tables – meaning there is not a separate table for each object type with separate columns for each individual property value of an object. Relationships between objects are maintained via the Relationship lookup table, and property data for each individual record is maintained within a memo field for each object.

Users will find it difficult to write SQL queries directly against these tables to build reports via ASP or JSP pages. It is recommended that users use our report generator tool or our COM (VBA) object model to access and report on objects in a System Architect repository.

The ENTITY Table


The ENTITY table contains a specification for each diagram, symbol, and definition that users have created in the encyclopedia. If you were to browse the **Entity** table using SQL Server 2000’s Enterprise Manager, it would look like the table shown in Figure 4, below.

Audit	Class	FromArrow	FromAssoc	ID	Name	SeqNum	Properties	ToArrow	ToAssoc	Type	UpdateDate	SA GUID
Suzanne	1	0	0	501	International Hotel Chain	0	[[[GUID]]]	0	0	80	2/12/2001 5:09:05 PM	44-84d1-00a0240f30d4
Suzanne	2	0	0	503	Booking	0		0	0	443	2/8/2001 5:43:07 PM	
Suzanne	2	0	0	502	Billing	0		0	0	443	2/8/2001 5:41:58 PM	
Suzanne	2	0	0	504	Customers	0		0	0	467	2/7/2001 6:17:25 PM	
Suzanne	2	0	0	505	Maintenance Dates	0		1	0	466	2/7/2001 6:19:51 PM	
Suzanne	2	0	0	506	Reservation Details	0		1	0	466	2/7/2001 6:21:57 PM	
Suzanne	2	0	0	507	Customer Details	0		1	0	466	2/7/2001 6:21:53 PM	
Suzanne	2	0	0	508	Renewal Dates	0		1	0	466	2/7/2001 6:21:48 PM	
Suzanne	2	0	0	509	Billing Details	0		1	0	466	2/7/2001 6:21:45 PM	
Suzanne	1	0	0	510	Hotel Booking System	0	[[[GUID]]]	0	0	81	2/12/2001 5:07:44 PM	59982bea-fcd7-11d4-84d1-00a
Suzanne	2	0	0	511	Marketing	0		0	0	471	2/7/2001 6:35:13 PM	
Suzanne	2	0	0	512	Management	0		0	0	471	2/7/2001 6:35:01 PM	
Suzanne	2	0	0	513	Customer	0		0	0	471	2/7/2001 6:34:50 PM	
Suzanne	2	0	0	514	Hotel Booking System	0		0	0	534	2/7/2001 6:35:11 PM	
Suzanne	2	0	0	515	Reservation	0		1	0	468	2/7/2001 6:36:12 PM	
Suzanne	2	0	0	516	Customer Analysis Details	0		1	0	468	2/7/2001 6:36:01 PM	
Suzanne	2	0	0	517	Report Request	0		1	0	468	2/7/2001 6:35:47 PM	
Suzanne	2	0	0	518	Reports	0		1	0	468	2/7/2001 6:35:36 PM	
Suzanne	2	0	0	519	Confirmation	0		1	0	468	2/7/2001 6:35:27 PM	
Suzanne	1	0	0	520	Booking System	0	[[[GUID]]]	0	0	82	2/12/2001 5:13:04 PM	59982bec-fcd7-11d4-84d1-00a
Suzanne	2	0	0	521	Reservation Billing	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	522	Reservation Cancellation	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	527	Existing Customer Maintenance	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	528	New Customer Entry	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	523	Reservation Creation	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	524	Reservation Maintenance	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	529	Customer Entry	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	525	Booking Reporting	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	526	Booking System	0		0	0	478	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	530		0		1	0	474	2/7/2001 6:53:36 PM	
Suzanne	2	0	0	531		0		1	0	474	2/7/2001 6:53:36 PM	
Comp	3	0	0	2121	Accounts Clerk	0	[[[GUID]]]	0	0	311	9/29/2002 3:23:54 PM	4d937afc-44ee-4825-923c-540
Comp	3	0	0	2122	Receptionist	0	[[[GUID]]]	0	0	311	9/29/2002 3:23:54 PM	d14e00d4-16ca-4713-88c8-18e

Figure 4. The ENTITY Table for a System Architect Encyclopedia

These artifacts – diagrams, symbols, and definitions – are the three ‘classes’ of artifacts that can be stored in an encyclopedia. A number is assigned for each class type: 1 = Diagram, 2 = Symbol and 3 = Definition. Each particular diagram, symbol, or definition that you create in an encyclopedia has its class type number listed next to it in the ENTITY table. In addition to this, each instance of a diagram, symbol, or definition is assigned a unique ID, a GUID, and other attributes such as Name, Audit, and UpdateDate. The ENTITY table is updated as you add or remove encyclopedia objects (or instances).

CROSS-REFERENCE Appendix B provides an explanation for each column specified in the ENTITY table.




The RELATIONSHIP Table

The RELATIONSHIP table creates a unique record for each *relationship* between encyclopedia objects. Figure 4 below shows an example of the RELATIONSHIP table.

ID1	ID2	Relation	Sequence
121	124	2	0
121	128	2	0
121	129	2	0
		35	
		15	
		26	
122	123		
122	121		0
123	124	21	0
123	122	27	0
124	121	3	0
124	122	14	
124	123	20	
126	121	21	
127	129	21	0
128	121		0
128	126		0
129	121	3	0
129	127	20	0
141	142	2	0
141	143	2	0
141	144	2	0
141	145	2	0
141	146	2	0
141	147	2	0
141	148	2	0
141	149	2	0
141	150	2	0
141	151	2	0
141	152	2	0
141	153	2	0
141	154	2	0
141	1001	35	0
142	141	3	0
142	149	8	0
142	151	8	0
142	153	8	0
142	659	20	0
143	141	3	0
143	149	6	0
143	152	6	0
143	654	20	0
144	141	3	0
144	152	8	0
144	664	20	0

Figure 4. RELATIONSHIP Table in System Architect Encyclopedia

Related objects are contained in columns ID1 and ID2, which use the object's ID number, as recorded in the ENTITY table. The relationship type is indicated by an ID number in the Relation column.

<p>CROSS-REFERENCE</p> 	<p>Appendix C provides an explanation of the relationship ID numbers used in the RELATIONSHIP table.</p>
---	--

Each relationship uses two records to describe it. For example, consider one relationship between a diagram named Hotel Reservation and a symbol on it named Customer. That one relationship is recorded in the table by two records – the first which reads, in effect, “diagram *contains* symbol”, and the second record which reads “symbol *is contained in* diagram”. In a way, a relationship is recorded from

each object's point of view, so that Object #1 *contains* Object #2 is one record, and Object #2 *is contained in* Object #1 is another record, even though both records describe the same relationship (see Figure 4, above).

The FILES Table

The FILES table contains a record for each ancillary file that is part of the encyclopedia. These are non-database files that are part of the encyclopedia database, but can be exported, edited with other applications, and then imported back into the database. This includes bitmaps and Windows metafiles (WMF) specified by users to be used on diagrams and toolbars, the configuration files that specify each encyclopedia's metamodel (saprops.cfg and usrprops.txt), the encyclopedia's stylesheet file (autoexec.sty), etc. For each file in the FILES table, the following attributes are recorded: Data, Date, Type, and Name (which contains the path and filename, with filename extension).

A Telelogic Working Paper

Data	Date	Name	Type
<Binary>	9/25/2002 3:09:23	AUTOEXEC.STY	<NULL>
<Binary>	9/19/2002 3:52:44	D0000001.DGX	<NULL>
<Binary>	9/19/2002 3:52:44	D0000001.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000002.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000002.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000003.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000003.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000004.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000004.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000005.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000005.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000006.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000006.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000007.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000007.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000008.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000008.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000009.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000009.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000010.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000010.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000011.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000011.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000012.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000012.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000013.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000013.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000014.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000014.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000015.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000015.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000016.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000016.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000017.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000017.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000018.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000018.wmf	<NULL>
<Binary>	9/23/2002 11:01:0	D0000019.DGX	<NULL>
<Binary>	9/23/2002 11:01:0	D0000019.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000020.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000020.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000021.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000021.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000022.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000022.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000023.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000023.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000024.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000024.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000025.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000025.wmf	<NULL>
<Binary>	9/23/2002 11:01:1	D0000026.DGX	<NULL>
<Binary>	9/23/2002 11:01:1	D0000026.wmf	<NULL>

Figure 5. FILES Table in System Architect Encyclopedia (part 1)

Data	Date	Name	Type
<Binary>	9/19/2002 1:56:33	images\slctent.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctent.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctfmpg.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctfmpg.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctfspg.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctfspg.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctint.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctint.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctjspg.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctjspg.wmf	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctmeta.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctscb.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctscb.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctserv.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctserv.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctsvpg.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctsvpg.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slcttggp.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slcttggp.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctwbpg.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctwbpg.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\slctwkr.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\slctwkr.wmf	<NULL>
<Binary>	9/19/2002 1:56:33	images\SLDIER12.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\SLDIER12.WMF	<NULL>
<Binary>	9/19/2002 1:56:33	images\TANK_12.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TANK_12.WMF	<NULL>
<Binary>	9/19/2002 1:56:33	images\Target.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\Target.WMF	<NULL>
<Binary>	9/19/2002 1:56:33	images\TargetHLCPTER4.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetHLCPTER4.WMF	<NULL>
<Binary>	9/19/2002 1:56:33	images\TargetPlane.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetPlane.WMF	<NULL>
<Binary>	9/19/2002 1:56:33	images\TargetPLNSILO8.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetPLNSILO8.WMF	<NULL>
<Binary>	9/19/2002 1:56:34	images\TargetSBMRINE1.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetSBMRINE1.WMF	<NULL>
<Binary>	9/19/2002 1:56:34	images\TargetSHIP_01.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetSHIP_01.WMF	<NULL>
<Binary>	9/19/2002 1:56:34	images\TargetSLDIER12.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetSLDIER12.WMF	<NULL>
<Binary>	9/19/2002 1:56:34	images\TargetTANK_12.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\TargetTANK_12.WMF	<NULL>
<Binary>	9/19/2002 1:56:34	images\WORLD_02.bmp	<NULL>
<Binary>	9/19/2002 1:56:32	images\WORLD_02.WMF	<NULL>
<Binary>	9/19/2002 1:56:32	images\XOR.wmf	<NULL>
<Binary>	9/19/2002 1:56:34	images\XOR_menu.bmp	<NULL>
<Binary>	9/19/2002 3:35:11	P0000001.WMF	<NULL>
<Binary>	9/19/2002 2:00:15	saderclar.cfg	<NULL>
<Binary>	9/26/2002 5:40:45	saprops.bin	<NULL>
<Binary>	9/19/2002 1:56:23	SAPROPS.CFG	<NULL>
<Binary>	9/19/2002 1:56:31	USRPROPS.TXT	<NULL>

Figure 6. FILES Table in System Architect Encyclopedia (part 2)

The FILES table contains the following files:

- **The Property Set Files:** Three property set files determine the metamodel, including the diagram and property sets that are turned on for an encyclopedia. They are as follows:
 - **SAPROPS.CFG** – created by Popkin Software. Specifies the metamodel of the encyclopedia.
 - **USRPROPS.TXT** – you add code to this file to specify metamodel extensions that you require.

- **SADECLAR.CFG** – specifies the diagram types and property sets that are turned on for the encyclopedia. You specify changes to this file using the Tools, Customize Method Support command in System Architect.
- **DGX and WMF Files Representing Each Diagram:** A .DGX and .WMF file pair is stored for each diagram in the encyclopedia.
 - ***.DGX Files** – DGX is the format for System Architect diagrams. It contains all graphical information about a diagram, including the symbol positions, color fills, text color, and so on. The file is automatically created when you create a diagram, and is named with the syntax Dnnnnnnn.DGX.
 - ***.WMF Files** -- By default, System Architect creates a .WMF file (metafile) for each .DGX file, and uses it to display a diagram thumbnail in the Browser's detail tab. If a diagram changes, its metafile is updated when the encyclopedia is re-opened. (This is different than the .WMF files that you can import and specify for symbols on diagrams, described below.)
- **Graphics that You Specify for Symbols on Diagrams and Toolbars:** System Architect enables you to specify new graphics for symbols drawn on diagrams in an encyclopedia. You specify that a symbol is depicted by a graphic that you supply using the Depictions clause in USRPROPS.TXT. Generally, you specify a Windows Meta File (.WMF) for symbols drawn on diagrams, and a corresponding Bitmap (.BMP) to represent that symbol on a toolbar.
 - ***.WMF Files** – You may specify your own graphics for symbols drawn on diagrams. You should use Windows Meta Files (.WMF) for symbols drawn on diagrams, because they are vector images that scale correctly when you drag on them to make them larger or smaller. You can use .BMP's as well, but they do not scale well. Please visit the Popkin Software website's Tip of the Week area (http://www.popkin.com/customers/customer_service_center/tip_of_the_week/tipofweekarchive.htm), within the section titled "Customization", for some pointers.
 - ***.BMP Files** – If you specify new graphics for symbols drawn on diagrams, you will want to represent those symbols in the toolbar as well. For the toolbar, using bitmap images is best, since there is no need for them to scale. Usually, it is best to create a 16x16 pixel bitmap for each symbol that you want to represent in the toolbar.

To specify your own graphics for symbols, you must make the necessary changes to USRPROPS.TXT, and import the respective .BMP and .WMF files that you are supplying into the FILES table of the encyclopedia.

Information on how to use the Depict command and specify your own graphics for symbols is provided in System Architect's help.

- **Graphics that You Paste or Copy Into an Encyclopedia:** A WMF or BMP file is created automatically every time you paste or copy a picture into a diagram. This is performed in System Architect by selecting Draw, Picture, File Copy or File Reference.
 - **File Copy:** If you choose File Copy, System Architect will ask you for the location of the file that you are copying onto the diagram. Once specified, System Architect will make a copy of the .BMP or .WMF file and place it into the FILES table, and paste that figure onto the diagram.
 - **File Reference:** If you choose File Reference, System Architect will ask you for the location of the file that you are copying onto the diagram. Once specified, System Architect will create a reference to the external folder where the file is located, and paste that figure onto the diagram. The full path of the referenced file is maintained in the .DGX file of the diagram.
- **The AUTOEXEC.CSV File:** When you start a new project, the AUTOEXEC.CSV file is automatically imported into the FILES table. AUTOEXEC.CSV contains the Trigger Templates, which can be used as the basis for defining triggers. Complete information on the use of triggers can be found in the on-line help for the Schema Generator.
- **The Format File (AUTOEXEC.STY, or 'style sheet'):** The format file determines the look of symbols that are drawn on diagrams. In particular, it specifies a symbol's size, shape, line thickness, font, text justification, etc. You can change these symbol characteristics to fit your project's needs.

It is not mandatory that the Format file be housed in the Files table. You may place the Format File as follows:

- You may put it in a central directory and then specify that path in the SA2001.ini file for each user. To do this you open the SA2001.ini file using a text editor such as Notepad (the sa2001.ini file is stored in the <C>:\WINNT directory), and then create a setting such as `FORMATFILE = <C>:\Program Files\Popkin Software\System Architect\Autoexec.sty`. In this case the name Autoexec.sty is arbitrary; you may name it something else, as long as the extension remains .sty (i.e., Payroll.sty).
- You may name the format file AUTOEXEC.STY and import a copy into the Files table of all your project encyclopedias (using System Architect's Tools, Encyclopedia File Manager command). The format file loaded in the FILES table of the encyclopedia is auto-loaded upon opening of the encyclopedia. It must be named Autoexec.sty

The PROPERTIES Table

The PROPERTIES table contains the VersionRestriction record, which primarily stores the version of System Architect used to create the encyclopedia database.

The IDGENERATOR Table

The IDGENERATOR table contains the ID number that will be assigned to the next object created in the encyclopedia. That object will be ENTITY table. For example, if

the last object created was assigned ID number 1000, the NextId column of the IDGENERATOR table will contain number 10001. Users should not touch this table.

The ERROR_LOG Table

The ERROR_LOG table contains the error log of all errors reported during operation of System Architect. This includes errors reported during most import/export operations of the tool, including reverse and forward engineering.

The INFO_LOG Table

The INFO_LOG table lists tests that have been run on the encyclopedia and their results. The file gets created in the encyclopedia being tested by such commands as Encyclopedia Verify and Repair.


The NETWORK_LOG Table

The NETWORK_LOG table contains all messages reported regarding network use of System Architect. These messages can sometimes prove useful in tuning a network. A Network Log option in the SA2001.ini file may be set to turn on or off network logging.

Creating or Opening an Encyclopedia

You create or open encyclopedias using System Architect's **Encyclopedia Open** dialog. You may create or open encyclopedias on your local instance of MSDE, or on the network (as long as SQL Server 2000 or MSDE has been installed and set up on the network for use with System Architect).


To create or open encyclopedias on the network, you must have sufficient access rights to SQL Server 2000 or MSDE, given to you by your System Administrator.

<p>CROSS-REFERENCE</p> 	<p>For detailed information on user access for V9 encyclopedias, please see the document, System Architect V9 Repository Access and Security, available on http://www.popkin.com/whitepapers.</p>
---	--

Creating an Encyclopedia

In summary, to create an encyclopedia, you do the following in the **New** tab of the **Open Encyclopedia** dialog:

1. Specify the server,
2. Create a database on that server, and

<p>IMPORTANT</p> 	<p>SA's encyclopedia open process MUST create the database, as it will build the encyclopedia's schema, and populate and enumerate the encyclopedia database as required for the tool to function properly. You cannot set up a database in SQL Server first and then have users use the designated database. Unless SA creates the database, SA users can not use it.</p>
---	---

3. Specify the directory in which System Architect will create a Universal Data Locator (UDL) file (which will provide a pointer to the location of the encyclopedia).

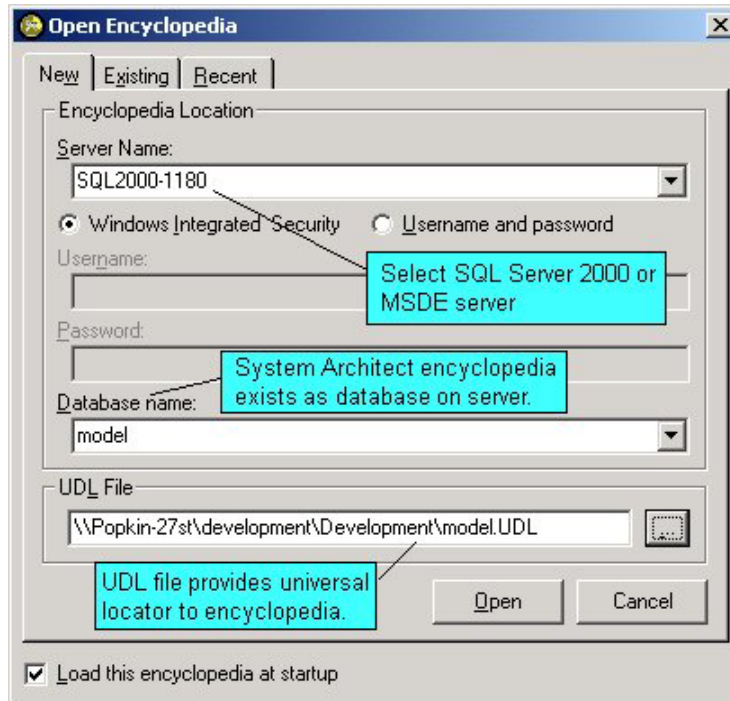


Figure 7. Creating a New Encyclopedia

Again, as stated above, you may specify that the UDL file get created in any directory. The UDL file is what you will use to access your encyclopedia in the future, so it is best to store it in a convenient location.

Step-by-step procedures on how to create an encyclopedia are provided in System Architect's on-line help.

Opening an Encyclopedia

Once created, opening an existing encyclopedia is performed via the **Existing** or **Recent** tabs of System Architect's **Open Encyclopedia** dialog. In either tab, you select the .udl file of the encyclopedia you wish to open.

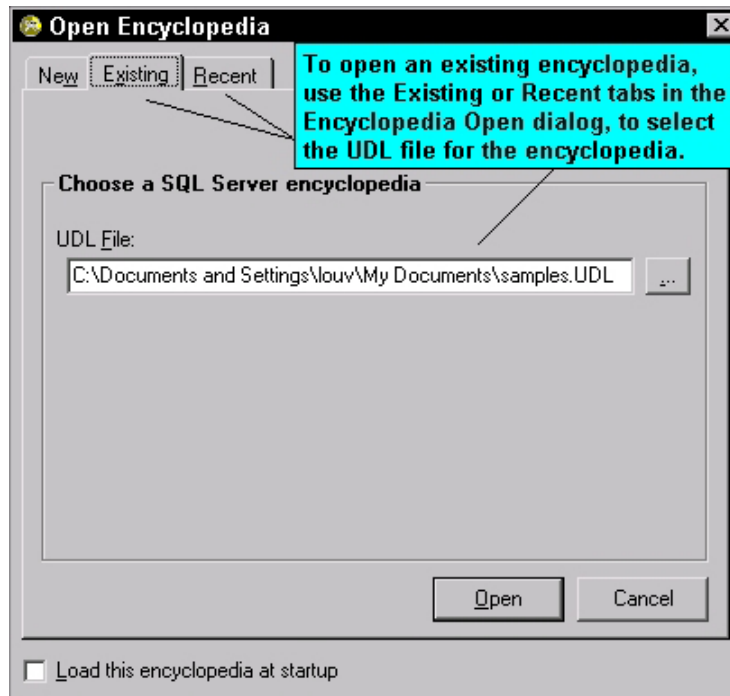


Figure 8. Opening an Existing Encyclopedia

Opening Prebuilt, Sample Encyclopedias Shipped With SA

System Architect provides a number of pre-built encyclopedias that you may use to browse example models already built, or to take the product's on-line tutorials available in the Help. The encyclopedias for the tutorials are partially built – they contain some models and definitions that are used as building blocks for taking the tutorial(s).

These encyclopedias are provided as MDF files located within subdirectories of the <C>:\Program Files\Popkin Software\System Architect\Encyclopedias directory. They include the following:

- **Samples.mdf** – a pre-built encyclopedia presenting models created for a sample Hotel Reservation business and its accompanying systems. This encyclopedia presents examples of most of the diagram types supported by System Architect.
- **Tutorial.mdf** – an encyclopedia that has some diagrams and definitions already created in it, intended as a building block for taking the on-line method tutorials available in System Architect's help.
- **Object Model.mdf** – contains one diagram – a UML class diagram that provides the model of System Architect's own COM (VBA) object model. You may use this diagram to reference classes and methods available in the object model, when building VBA macros for System Architect.
- **Stock Control.mdf** – an encyclopedia that has some diagrams and definitions already created in it; intended as the building block for

taking the on-line simulation tutorial available in the help. You must have the Simulation option to System Architect to take this tutorial.

- **C4ISR Samples.mdf** – a pre-built encyclopedia presenting example models for the C4ISR Framework option to System Architect.
- **Magic Sample.mdf** – a pre-built encyclopedia presenting an example diagram pre-populated with definitions for System Architect's interface to Magic, the 4GL tool.

System Architect's Start-Up wizard enables you to attach these encyclopedias to your server before you begin work. By default, all encyclopedias are selected to be attached. If you have deselected any to be attached in the wizard, you can attach them later by rerunning the Start-Up Wizard, or by using Popkin Software's **System Architect Encyclopedia Manager (SAEM)** or Microsoft's Enterprise Manager. (Enterprise Manager is available from the SQL Server 2000 Standard Edition CD-ROM.)

Moving or Making Copies of Encyclopedias

There are times when you may want to send a copy of an encyclopedia to another user. To send an encyclopedia to someone, you have two options:

- Make a full backup of the encyclopedia using SAEM or Microsoft's Enterprise Manager. This creates a .BAK file that you may send to another user. It is recommended that you compress this file before sending, as .BAK files can be quite large. The receiver of this file may use SAEM or Microsoft's Enterprise Manager to restore the .BAK file, which in effect creates a copy of the encyclopedia on their server. The advantage of this approach is that the original encyclopedia is never detached from its server, so no users ever have to stop working.
- Use SAEM or Microsoft's Enterprise Manager to unattach the database from its server, locate the unattached .MDF file, and send it to another user. It is recommended that you compress this file before sending, as .MDF files can be quite large. At the other end, the receiver of the .MDF file must attach it to his or her server before using it, and then create a UDL file for it in System Architect. This is not an ideal way to send a network copy of the encyclopedia to someone, because it means that at some point, you have to unattach the database from the server, which in a network environment means users have to stop work for a short while.

Option 1 (Preferable): Making and Sending a Backup (.BAK) File

The steps to make and send a backup file of your encyclopedia to send to another user using SAEM are provided below.

1. Connect to your server in SAEM.
2. Select the database you wish to send by choosing **Database, Select Database**.
3. Select **Database, Backup**.
4. In the **Name** field of SAEM's **Backup Database** dialog, specify a name for the backup database. By default, the name of the database is the original name with '_backup' amended to it.
5. Specify a destination where the .BAK file will be placed.
6. Make sure that Backup type is set to **Database – complete**.
7. Click the **Backup** button. The .BAK file is created in the directory that you specified.
8. Compress (or zip up) the .BAK file and email it or make it available to your intended recipient.
9. The recipient unzips the .BAK file, and 'restores' it on their server. He or she selects **Database, Restore** in SAEM.
10. In SAEM's **Database Restore** dialog, specify a name for the new database in the **Restore database as** field.
11. In the **Restore** from group, toggle on **Backup** file.
12. Use the browse button to find and select the .BAK file. Select **Restore**.

Option 2: Unattaching and Sending the .MDF File

The steps to unattach and send the encyclopedia as an .MDF file are as follows:

1. Unattach the encyclopedia from the server it is attached to using SAEM (for MSDE) or SQL Server's Enterprise Manager (for SQL Server 2000). Once you unattach the encyclopedia, which is an MDF file, you may move or copy the file from the directory it is in.

WARNING



SAEM may only be used to attach and detach files from the machine running MSDE. You may not attach or detach from the a remote machine unless you are using Enterprise Manager from SQL Server 2000.

2. Locate the MDF file. This file is by default located in SQL Server's DATA directory. This MDF file is stored, by default, in SQL Server's Data directory. If you are running a local instance of MSDE, this directory is typically in your <C>:\Program Files\Microsoft SQL Server\MSSQL\$POPKINSQL\Data directory.
3. Make a copy of the MDF file and place it in a directory from which you will send it. Reattach the original MDF file to its server so that users can continue working.

WARNING



Don't rename the MDF file at this stage. If you rename it, you may have trouble trying to reattach it to the server or another server later.

4. Take the copy you created of the MDF file, zip it up, and email it or make it available to your intended recipient.
5. The recipient must attach the MDF file to their server using SAEM or SQL Server's Enterprise Manager (for SQL Server 2000) before using it.
6. The recipient must create a UDL file for the encyclopedia via System Architect's Encyclopedia Open dialog.

Customizing an Encyclopedia

System Architect's repository environment features a renowned ability to easily extend the underlying metamodel of information stored, enabling you to tailor the tool to meet the exact modeling needs for your project. You may create new diagram

types, symbol types, and definition types, and add new properties to new or existing definitions.

Besides powerful metamodel changes, you can also change the look of the information that you are modeling by specifying new pictures for how symbols are depicted on diagrams. You may provide your own bitmaps and metafiles or use pictures provided by Popkin Software on its website (http://www.popkin.com/customers/customer_service_center/addins/images/clipartlib.htm).

None of these features have changed in System Architect V9. However, the files used to customize the encyclopedia must now be moved in and out of the FILES table of the database. The next section describes how that is done.

TIP



The key difference between working with USRPROPS.TXT in System Architect V9 is that the USRPROPS.TXT file is located in the Files table of the SQL Server database. You must export it out of the Files table of the database, place it in any directory, edit it, and then import it back into the database. Exporting and importing the USRPROPS.TXT file is accomplished using either the File Manager utility in System Architect (Tools menu), or the Encyclopedia File Manager (Tools menu).

Modifying the Encyclopedia's Metamodel

You extend the encyclopedia's underlying metamodel by adding code to the User Properties (USRPROPS.TXT) file. Every time an encyclopedia is opened, System Architect parses the encyclopedia's saprops.cfg file, which contains the default metamodel specified by Popkin Software, and then it parses the encyclopedia's usrprops.txt file, which supercedes and extends the code in saprops.cfg.

In System Architect V9, Both the saprops.cfg and usrprops.txt files are contained in the FILES table of the encyclopedia database. To edit the Usrprops.txt file, you must perform the following steps:

Export the Usrprops.txt File from the Encyclopedia's FILES Table

1. Select Tools, Customize User Properties, Export USRPROPS.TXT (Encyclopedia).
2. In the Export User Properties dialog, select a directory to export the file to and click OK. The file will be placed in the specified directory, and automatically opened in Notepad.
3. Edit the file as appropriate. (For more information about editing the USRPROPS.TXT file, refer to the System Architect Extensibility Guide, located in <C>:\Program Files\Popkin Software\System Architect\Manuals\Extensibility_USRPROPS.pdf.)

Import the Usrprops.txt File Back Into the Encyclopedia's FILES Table

4. Select Tools, Customize User Properties, Import USRPROPS.TXT (Encyclopedia) to import the file back into the Files table of the encyclopedia.

5. Reopen the encyclopedia (File, Encyclopedia Open command) to make your changes take effect.

Specifying Your Own Images for Symbols on Diagrams

System Architect enables you to specify your own graphics for symbols drawn on diagrams in an encyclopedia. You specify that a symbol is depicted by a graphic that you supply using the Depictions clause in USRPROPS.TXT.

Specifying Your Own Images for Existing Encyclopedias

To specify your own images for symbols, you must do the following:

1. **Make the necessary changes to USRPROPS.TXT.** You use the DEPICTIONS command (and, optionally, the RETAIN STYLE command). Information on how to make the necessary code changes are provided in System Architect's help. An example for such code is:
List "Node Stereotypes"
{
Value "Firewall" depictions {diagram images\firewall.wmf menu images\firewall.bmp}
}
2. **Import your .BMP and .WMF files into the encyclopedia's FILES table.** You may either use System Architect's Encyclopedia File Manager (Tools, Encyclopedia File Manager), or SAEM, or Microsoft's Enterprise Manager to import your user-defined graphics files into the FILES table of the encyclopedia database. Encyclopedia File Manager can only import one file at a time. If you have multiple graphics files, we recommend you use SAEM to import the files into the FILES table. The names of the files that you import should be consistent with your Usrprops.txt code. You should append the name of the file with 'images/' to simulate that it is in an 'images' subdirectory of the FILES table. If you use SAEM to import multiple files at a time, make sure that they are in a directory that is named 'images', located anywhere on your computer. SAEM will automatically append the name of all files imported from a directory named images with 'images/' at the front of each graphic's file name. The reasoning for this is explained in the tip below.

TIP



In versions of System Architect prior to V9, user-specified images were placed in the Images subdirectory of each encyclopedia. In V9, there is no such Images folder in the FILES table of an encyclopedia/database. However, the same convention should still be used – you should precede your user-specified images with 'images\' appended to the front of their name.

There are two advantages to using this strategy. First, it provides a sort of name independence and logical grouping strategy for user-specified images. Second, it is consistent with the way that images are handled when new encyclopedias are created – System Architect takes all graphics in the ..\System Architect\images directory, places them in the FILES table of the new encyclopedia, and gives them a name that is appended with 'images\'.

Explanation: In the code in step 1 above, the user-provided graphics are specified with the relative path statement `images\`. This is an images directory relative to the encyclopedia path. In versions of System Architect prior to V9, this was physically true; in System Architect V9 this is no longer the case – there is no images subdirectory inside the FILES table of the database.

TIP



If you are creating a new encyclopedia, you have an option – you can create the encyclopedia first and then import one or more user-provided graphics files into it via SAEM (SAEM V9.1 and later supports multiple-file import), or you may place your user-provided images into System Architect's main images directory (under the main software directory – System Architect\images) before creating the encyclopedia. System Architect takes all graphics in its main images directory and places them in the Files table of all new encyclopedias created.

The files that you import should meet with the following specifications:

- ***.WMF Files** – You may specify your own graphics for symbols **drawn** on diagrams. You should use Windows Meta Files (.WMF) for symbols drawn on diagrams, because they are vector images that scale correctly when you drag on them to make them larger or smaller. You can also use .BMP's , but they do not scale well.
- ***.BMP Files** – If you specify new graphics for symbols drawn on diagrams, you will want to represent those symbols **in the toolbar** as well. For the toolbar, using bitmap images is best, since there is no need for them to scale. Usually, it is best to create a 16x16 pixel bitmap for each symbol that you want to represent in the toolbar.

3. Reopen the Encyclopedia for the changes to take effect.

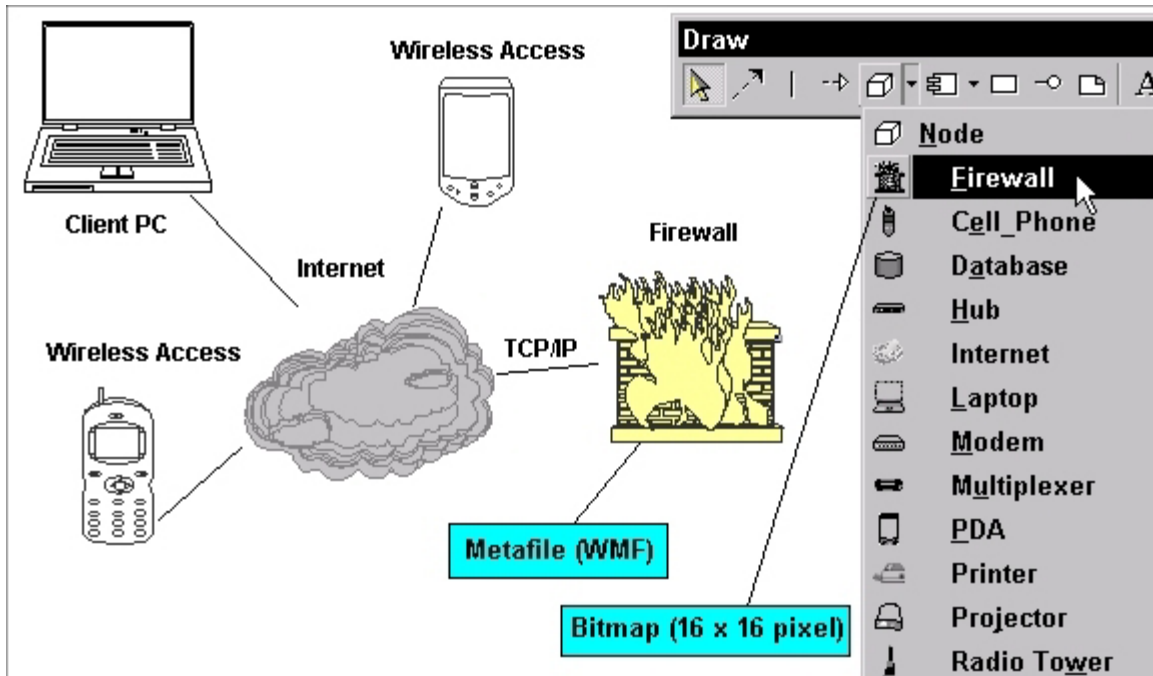


Figure 9. Using Your Own Custom Images

Specifying Your Own Images for New Encyclopedias

If you wish the same user-specified graphics files to go into all new encyclopedias that you or other team members create, perform the following steps:

1. **Copy and Paste Your .BMP and .WMF files into System Architect's 'Images' Subdirectory.** Before creating new encyclopedias, place your .BMP and .WMF files into the Images directory within the System Architect main program directory. All team users that will be creating new encyclopedias at any time in the future should do this. These files will automatically be placed in the FILES table of the encyclopedia that are later created. System Architect will append each file name with 'images\'', so a figure called Fred.bmp will be created in the new encyclopedia's FILES table with the name images\Fred.bmp. This is a shortcut to creating the encyclopedia, and then importing the user-provided graphic files into the encyclopedia afterwards.
2. **Make the necessary changes to USRPROPS.TXT.** You use the DEPICTIONS command (and, optionally, the RETAIN STYLE command). Information on how to make the necessary code changes are provided in System Architect's help. An example for such code is:

```
List "Node Stereotypes"
{
Value "Firewall" depictions {diagram images\firewall.wmf menu images\firewall.bmp}
}

```
3. **Reopen the Encyclopedia for the changes to take effect.**

Copying and Pasting Symbols Onto Diagrams

You may paste external graphics onto diagrams in System Architect. The graphics that you paste may be copied into the encyclopedia, or you may leave the graphic in a directory external to System Architect, and reference it.

Picture File Copy

Copying a picture from an external directory onto a System Architect diagram (Draw, Picture, File Copy) creates a copy of the bitmap (.bmp) or metafile (.wmf) specified, and places that copy into the **Files** table of the encyclopedia database. The name of the picture copy is provided by System Architect, with a naming convention of Pnnnnnnn.bmp or Pnnnnnnn.wmf (where nnnnnnn = a sequential 7-digit number).

Picture File Reference

You may also create a reference to a picture in an external directory, as you could with previous versions of System Architect (Draw, Picture, File Reference). System Architect creates a reference to the bitmap or metafile specified (it **does not** create a copy in the **Files** table of the encyclopedia database). If the picture is removed or moved from the external directory where it resides, the image of it on the System Architect diagram will disappear. The full path of the referenced file is maintained in the .DGX file of the diagram.

Maintaining An Encyclopedia

To maintain your encyclopedia, the following actions should be performed on a regular, scheduled basis by the Systems Administrator.

1. Backup the Encyclopedia – Use SAEM or Microsoft's Enterprise Manager. This creates a .BAK file of your database. You can version these backups in any popular version-control software.

2. Shrink the Encyclopedia – Ensure that there are no users connected to your database and shrink it using Microsoft SQL Server's Enterprise Manager tool, or Popkin Software's SAEM (System Architect Encyclopedia Manager) tool for MSDE. This step will get rid of artifacts that have been marked for deletion in the encyclopedia, and make the encyclopedia smaller. Instructions on how to shrink a database are provided in the on-line help of each respective tool. (This is the equivalent of the Encyclopedia Reorganize command available in System Architect versions V8 and earlier.)

3. Run Dictionary Update – Open the encyclopedia in System Architect and run the Dictionary Update facility on an encyclopedia by selecting it from the Tools menu. It is a good idea to ensure that no other users are working in the encyclopedia. This function does the following:

- Searches the dictionary for all undefined data elements and structures.
- Searches the encyclopedia for all dictionary entities without a correct relationship to the entity of which they are a part. For example, it searches for data elements whose relationships to data structures are not correctly defined. As much as possible, it will correct the errors following the parameters of the Property Configuration Set and the **USRPROPS.TXT** file.
- Searches the dictionary for undefined data or expressions.
- Creates and displays a report listing rules violations that could not be fixed.

4. Run Verify and Repair – Open the encyclopedia in System Architect and run the Verify and Repair facility on an encyclopedia by selecting it from the Tools menu. It is a good idea to ensure that no other users are working in the encyclopedia. This utility passes through all diagrams, symbols and definitions in the encyclopedia, and if it finds no errors, stops. If it finds errors, it takes another pass through the encyclopedia to fix these errors. It will automatically take a maximum of 3 passes to correct errors and then stop (or stop at any point where it achieves zero errors). Errors are reported to the screen and also written to the Verify.Log in the FILES table in the encyclopedia's database. This log file is appended with each run of Verify and Repair. The ERROR_LOG table records date, time and error counts of each Verify and Repair run.

When Verify and Repair finishes running, you should review the number of errors. Typically you should see zero errors. If there are errors, you should continue to perform Verify and Repair until you get no more than 10 errors. Should you come to a point where you have more than 10 errors and you perform Verify and Repair more than twice and the number of errors does not decrease any further, you should call [Technical Support](#).

Appendix A: Attaching and Opening an Encyclopedia

In this section, we provide detailed steps for attaching and opening encyclopedias.

Overall Steps

To open an encyclopedia, it must first be attached to your server. You must then create a UDL file for the attached encyclopedia. The overall steps are as follows:

1. Attach the encyclopedia (database .MDF File) to the server.
2. Specify the directory in which System Architect will create a Universal Data Locator (UDL) file (which will provide a pointer to the location of the encyclopedia).

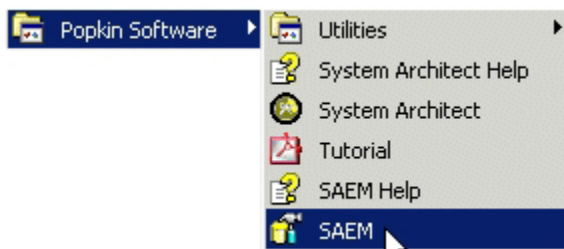
Attaching an Encyclopedia to Your Server

You attach an encyclopedia (its MDF file) to your SQL Server 2000 server or your MSDE server using the following tools:

- **For SQL Server:** You attach the database (MDF file) to your server using SQL Server's Enterprise Manager.
- **For MSDE:** Attach the prebuilt encyclopedia (MDF file) to your server using Popkin's **SAEM** (System Architect Encyclopedia Manager) tool.

For example, to attach an encyclopedia (.MDF file) to a MSDE server using SAEM perform the following steps:

1. Run the SAEM tool by selecting Start, Programs, Popkin Software, SAEM from your Windows status bar.



2. In the **Available SQL Servers** drop-down, select the server that you wish to attach the encyclopedia to -- this is the server that you will be working on. If you have installed MSDE to your local machine, you should see the name of your machine in the drop-down list. If you are working on the network, you should see available network servers in the drop-down list.
3. Select a method for **Authentication**, as follows:
 - If you are using a Windows NT-based operating system (Windows 2000, Windows XP, Windows NT, etc), you may use either Windows NT authentication or SQL Server authentication, as follows:

1. Toggle on **Windows NT** authentication, or
2. Toggle on **SQL Server**, and in the **Login Information** group, type in a Username of **sa** (for System Administrator), and leave the Password blank. (Note: if you are using a server on the network, you should use the username and password that your system administrator provides to you.)
 - If you are using Windows 98 or Windows ME, you must use SQL Server authentication. Toggle on **SQL Server**, and in the **Login Information** group, type in a Username of **sa** (for System Administrator), and leave the Password blank. (Note: if you are using a server on the network, you should use the username and password that your system administrator provides to you.)
4. Click **Login**. The **Encyclopedia Manager** main dialog should be presented.
5. In the **Encyclopedia Manager** main dialog, select **Attach** and click the **Execute** button.
6. In the **Attach a Database** dialog, click the ... (browse) button, and navigate to your database (.MDF file stored in a directory). Select the .mdf file for your database and click **Open**.
7. Back in the **Attach a Database** dialog, in the **Attach as** text box, type in the name of your database.
8. Click the **Attach** button to attach your encyclopedia to your server.
9. Click **Logoff** and then **Exit** to close **SAEM**.

Create a UDL File for the Encyclopedia

Once the encyclopedia (.MDF file) is attached to the server, you must create a UDL file for it. This is performed in System Architect as follows:

1. Run System Architect from the **Start, Programs** menu or double-click on the **System Architect** program icon
2. From the **File** menu in System Architect, select **Open Encyclopedia**.
3. On the **New** tab of the **Open a Project** dialog, specify the name of your server in the Server drop-down list.
4. Select a method for **Authentication**, as follows:

For SQL Server on the Network:

Select SQL Server, and fill in the Username and Password that your system administrator has provided to enable you to access the network server.

CROSS-REFERENCE



For detailed information on user access for V9 encyclopedias, please see the document, **System Architect V9 Repository Access and Security**, available on <http://www.popkin.com/whitepapers>.

Access Rights for SQL Server 2000:

Server Role: For users to be able to create an encyclopedia on a SQL Server 2000 server, they must either have a Server Role of System

Administrators or Database Creators. This is granted to the user by the System Administrator through SQL Server 2000 Enterprise Manager.

Database Access: If users are not give server roles, it will be necessary to assign users login permissions to specific databases and assign the following database roles for user to be able to open an encyclopedia and read and create definitions within it

- **db_datareader**
- **db_datawriter.**

This is granted to the user by the System Administrator through SQL Server 2000 Enterprise Manager.

Users must also have EXECUTE privileges to the following stored procedures that SA created in each database:

- EntityExistsById
- GetNextId
- LockEntityById

System Architect also makes use of the 'xp_userlock' Extended Stored Procedure with the **master** database. User's will need EXECUTE permissions on this procedure. It is sufficient to give the public role, which is already assigned to this procedure, EXECUTE permissions. This negates the need to add the user's login as a login in the **master** database.

For MSDE on your Local Machine:

- a. If you are using a Windows NT-based operating system (Windows 2000, Windows XP, Windows NT, etc), you may use either Windows NT authentication or SQL Server authentication, as follows:
 - Toggle on **Windows NT** authentication, or
 - Toggle on **SQL Server**, and in the **Login Information** group, type in a Username of **sa** (for System Administrator), and leave the Password blank.
 - b. If you are using Windows 98 or Windows ME, you must use SQL Server authentication. Toggle on **SQL Server**, and in the **Login Information** group, type in a Username of **sa** (for System Administrator), and leave the Password blank.
5. From the **Database Name** drop-down list, select your database.
 6. By the **UDL File** property of the **Open Encyclopedia** dialog, click the browse button. Specify a directory location and UDL file name – the name should be the same as the name of the database. The path that you place the UDL file is up to you -- for example, you may create it in your **My Documents** folder, or in the <C>:\Program Files\Popkin Software\System Architect\Encyclopedias directory, as shown below. The name does not have to be the same as the database, but for clarity's sake, it helps.
 7. If not already selected, toggle on **Load this project at startup**, located at the bottom left of the **Open a Project** dialog. This will make this encyclopedia the default encyclopedia that will automatically open each time you start SA.
 8. Select **Open** to open the encyclopedia.
 9. If you receive the message "**An encyclopedia already exists at this location. Would you like to open this encyclopedia?**", click **OK**. You will open the encyclopedia and create the new UDL file, which is a

reference to it. A database may have more than one UDL file associated with it.

Appendix B: ENTITY Table Contents

This appendix describes the contents of the ENTITY table in a System Architect encyclopedia. The ENTITY table contains a specification for each diagram, symbol, and definition that users have created in the encyclopedia.


<p>CROSS-REFERENCE</p> 	<p>The ENTITY table is introduced in the section of this paper titled Going Deeper – The Encyclopedia’s Internal Tables.</p>
---	---

Table 1. The ENTITY Table of a System Architect Encyclopedia

ENTITY Column	Explanation
Name:	Is an 80-character user- assigned name. This is the name of the diagram, or the symbol, or the definition that the user has created.
ID:	Is a 32-bit system-assigned identification number. This is a number unique in the encyclopedia. This is sometimes called the DDID.
Class:	Is the class code – 1 stands for diagram, 2 stands for symbol, 3 stands for definition.
Type:	Is the type code.
Number:	Is the symbol's number, for example, it is the <i>1</i> in <i>Process P1</i> or in <i>Process 2.3.1</i> .
FromArrow, ToArrow:	Is a True/False (0/1) indicator for the presence of an arrowhead at the from-end (FromArrow) or to-end (ToArrow) of a line symbol.
FromAssc, ToAssoc:	Is a code indicating what kind of multiplicity notation is to be drawn at the from-end and to-end of a line symbol. <ul style="list-style-type: none"> • Unmarked (code = 0) • One and Only One (code = 1) • Zero or One (code = 2) • One or Many (code = 3) • Zero, One or Many (code = 4) • Many (code = 5) • Unknown (code = 6) • Unmarked (code = 7) • Super (code = 8) • Sub (code = 8)
UpdateDate:	Is a system-assigned entry or last update date. In a network environment, it may also contain the date when an item was last locked.
UpdateTime:	Is a system-assigned entry or last update time. The time is represented in the 24-hour format HHMMSS. In a network environment, it may also contain the time when an item was last locked.
Audit:	Is the user-entered entry or last update audit identification. In a network environment, it may also contain the Id of the user who last locked the item.
SaGuid:	Is a 32-bit, system-assigned identification number which is unique across

A Telelogic Working Paper

	encyclopedias.
SeqNum:	Is the symbol's number: for example, the 1 in Process P1 or in Process 2.3.1.
Properties:	Is a pointer to this entry's memo field in the ENTITY table, if such a memo field exists. The memo field is where lengthy text is kept, such as Descriptions and other properties.
GUID:	Is a system-assigned identification number that is, in theory, unique across the universe.

Appendix C: Relationship Numbers

The table below lists the relationship numbers between objects in a System Architect encyclopedia. An example, relation number 2 is one of the relation numbers that describes the relationship between a diagram (UML Class diagram) and a symbol (class symbol called Customer, say), because the diagram **contains** the symbol. Relationship #1 is not used in System Architect – the relationship numbers start at #2 (see table below).

CROSS-REFERENCE



Use of relationship numbers is described in the section of this paper titled **Going Deeper – The Encyclopedia’s Internal Tables**.

Relation Number	Relation	Relationship Between Objects
2	contains	a diagram contains a symbol
3	contained in	a symbol is contained in a diagram
4	expands to	a symbol expands to a diagram
5	expands from	a diagram expands from a symbol
6	connects	a symbol connects to the start of a line
7	connects	the start of a line connects to a symbol
8	connects	a symbol connects to the end of a line
9	connects	the end of a line connects to a symbol
10	connects	a module connects to and sends data via a flag symbol
11	connects	a flag symbol connects to and receives data from a module
12	connects	a module connects to and receives data from a flag symbol
13	connects	a flag symbol connects to and provides data to a module
14	uses	an expression uses data (elements or structures)
15	used by	data (elements or structures) are used by an expression
16	explained by	a symbol is explained by a comment
17	explains	a comment explains a symbol
18	addresses	a symbol addresses a requirement, test plan, etc.
19	addressed by	a requirement, test plan, etc. is addressed by a symbol
20	defined by	a symbol is defined by a definition
21	defines	a definition defines a symbol
22	qualified by	a line (or rectangular) symbol is "qualified by" a flag symbol
23	qualifies	a flag symbol "qualifies" a line symbol (or rectangular symbol). This relation is not currently used in System Architect.

24	is a	a definition "is an instance" of a definition
25	instantiated by	a definition is "instantiated by" a definition
26	identifies	an object "identifies" another object
27	is keyed by	an object has another object as part of its key. For example, a column is keyed by a table; a table identifies a column.
28	embeds	a symbol wholly embeds another symbol (e.g., use case diagram)
40	is child of	a symbol is the child of another symbol (e.g., decomposition diagram)
41	is parent of	a symbol is the parent of another symbol
42	is first child of	a symbol is the first child of another symbol (e.g., decomposition diagram)
43	has first child	a symbol has another symbol as first child
44	is next sibling of	a symbol is the next sibling of another symbol (e.g., decomposition diagram)
45	is prior sibling of	a symbol is the prior sibling of another symbol
48	originated from	an object originated from a definition (e.g., graphic screen diagram)
49	is origin of	a definition is origin of derived object
50	is based on	an object is based on a definition (usually a data element)
51	is basis for	a definition is basis for derived object