Rational. System Architect

IBM

Conversion Guide

IBM Rational

System Architect Conversion Guide

Release 11.3.1

Before using this information, read the "Notices" in the Appendix, on page 6-2.

Table of Contents

Table of Contents

ii

1

Introduction to Conversion

Introduction

This chapter provides an overview of conversions necessary to bring work created in earlier Rational System Architect versions to its latest version. This includes moving up to V10.4 and later and from previous versions (V10, V9, V8) moving to V8 from v6 and v7, and up to v6 or v7 from versions 4.0 and/or 3.1.

Overview of the Conversions

System Architect V11.3.1 automatically upgrades encyclopedias created in System Architect V10.7 and later to V11.3. System Architect V10.3 and later creates new stored procedures and tables (for history) and makes changes to existing stored procedures in all encyclopedias. Because of this, encyclopedias created in V10.1, V10.0, or V9 of the product must first be converted/upgraded for use in System Architect V10.3 format before they can be upgraded to V11.0. The instructions on how to do this follow in the next section.

If the version of System Architect that you are using is V9 or V10.0, and you have created custom macros that run off menu items, you will need to adjust them slightly for use in SA V10.1 and later. Details are provided in Chapter 3 of this manual.

Opening a V9 encyclopedia with V10 requires the System Architect administrator to grant access rights to a new GetFileSize stored procedure. Details are provided later in this chapter.

The conversion of V8 encyclopedias to V9, V10, or later is accomplished by following the procedures in Chapter 2. To convert an encyclopedia built with a previous version of System Architect to V9 (SQL Server underlying database), the encyclopedia must be converted to V8 first.

The conversions for previous versions of the product to V8 are performed by invoking the conversion routine, which is accessed in System Architect by selecting **Tools, Conversion** (with any encyclopedia open). If you are converting an encyclopedia created with a previous version of System Architect (versions 6 or 7), a GUID/UML conversion will run and create a new, converted encyclopedia in a new path (the old encyclopedia remains intact).

If you are converting an encyclopedia created with a System Architect 4.0 product, a data-modeling/long-names conversion will run first to convert the encyclopedia to System Architect status, and the GUID/UML conversion will run

second to take the resulting, converted encyclopedia and convert it to System Architect V8 status.

Encyclopedia Schema Overview

Rational System Architect provides encyclopedia options, that to become enabled, require encyclopedia schemas to be changed. The sections below describe the Standard Schema, the Workspace Schema, and the Optimized XT Reader feature.

Standard Schema

A Standard Schema is one that has not been enabled to use Workspaces. By default, new encyclopedias are created with the Standard Schema. Furthermore, new encyclopedias on SQL Server 2005 or SQL Server 2008 are created with the Standard Schema, and are enabled with the Optimized XT Reader feature, described further down.

Workspaces Schema

The Workspaces Schema enables you to baseline an encyclopedia and create Workspaces for it. A baseline encyclopedia becomes read-only. For information on working with Workspaces, see Manage Workspaces Overview within the SAEM for SQL Server online help.

Optimized XT Reader

Encyclopedias enabled with the Optimized XT Reader feature enables Rational System Architect to integrate with Rational Publishing Engine Document Studio by making encyclopedias available as data sources for that product. This is made possible through a REST API. This feature also increases the number of concurrent users that can login to Rational System Architect XT with the 'Reader' mode.

By default, new encyclopedias you create on SQL Server 2005 or SQL Server 2008 are enabled with the Optimized XT Reader feature. SQL Server 2005 encyclopedias created with Rational System Architect versions prior to V11.3.1 must be converted before they can be opened. A dialog informs you of this requirement and prompts you to enable the Optimized XT Reader. Choosing 'Yes' converts the encyclopedia, while choosing 'No' leaves the encyclopedia intact, but it will not be opened.

Converting SA V 11.0 Encyclopedias with the UML 2 Configuration Property Set to SA V11.1

Upgrading UML 2 Encyclopedias from SA 11.0 to SA 11.1

If you created an encyclopedia with SA V 11.0 and enabled the UML 2 property set, you need to run the 'SAEncyclopediaConversion.mac' macro to upgrade that encyclopedia for SA V 11.1, as follows:

1. If System Architect running, close and reopen it.

2. Click **File** and select **Open Encyclopedia**.

3. Select the encyclopedia you need to convert and click **OK**.

4. A dialog message informs you that the encyclopedia is configured for an earlier version of System Architect, and asks if you want to upgrade it. Click **No**.

5. Click on **Tools** and select **Macro Projects**.

6. In the **Macro Projects** dialog, click the **Add Macro** icon.

7. Navigate to the subfolder <INSTALLATION DIRECTORY>\SAUML2\1.1\etc\, select the **SAEncyclopediaConversion.mac** macro and click **Open**.

8. Click **OK** to close the **Macro Projects** dialog.

9. A Microsoft Visual Basic dialog informs you "macros may contain viruses." Click **Enable Macros**.

10. Click on **Tools** and select **Run Macro**.

11. Select the **ExportForConversion11_0** macro
    and click **Run**. The **Macros** dialog is closed.

12. Close and reopen System Architect.

13. Reopen the encyclopedia you are converting.

14. A dialog message informs you the encyclopedia
    is configured for an earlier version of System
    Architect, and asks if you want to upgrade it.
    Click **Yes**.

15. Click on **Tools** and select **Run Macro**.

16. Select **ImportForConversionFrom11_0** and
    click **Run**.

Upgrading to SA V10.4 and later – Converting SA V10.3, V10.1, V10.0, and V9 Encyclopedias

Conversion Procedure to Upgrade Encyclopedias to SA V10.4 and later

Beginning with System Architect V10.3 new stored procedures and tables (for history) are created and changes are made to existing stored procedures in all encyclopedias (SQL Server 2000, MSDE, SQL Server 2005, SQL Express, Oracle 9i or Oracle 10g databases). To upgrade System Architect V10.4 and later encyclopedias to V11.0 you only need to open the V10.4 and later encyclopedias with System Architect V11.0.

However, encyclopedias created with V10.1, V10.0, or V9 of the product must be converted to V10.3 format before it can be updated automatically to V11.0. This is a two-stage process:

Note: SAEM cannot perform standard functions on an encyclopedia residing on a remote server.

Stage 1 – convert the data of the encyclopedia using SAEM, as follows:

1. Run System Architect Encyclopedia Manager (SAEM) and select **Start, Programs, Telelogic, Telelogic Lifecycle Solutions, System Architect, SAEM (SQL Server**) or **SAEM (Oracle**), depending on your server.

2. In SAEM, log in to the server containing the encyclopedia you wish to upgrade (select **Server, Connect**).

3. Choose the encyclopedia you wish to upgrade (select **Database, Select Database**).

4. In SAEM, select **Tools, Convert to 10.3**. This will convert the data to the format necessary to be used with System Architect 10.3.

Stage 2 – A person with a System Administrator's role, or the owner of the encyclopedia, must open the converted encyclopedia in System Architect V10.3 or later, as follows:

1. Find out whether the owner of an encyclopedia is using SAEM (for MSDE) or Microsoft's Enterprise Manager (for SQL Server). Have that person or a person with a System Administrator's role, open the converted encyclopedia in SA V10.3 or later.

2. System Architect will automatically try to add new stored procedures and tables to the existing encyclopedia when it is initially opened with SA V10.3 or later. As long as a person with a Systems Administrator role or the owner of the encyclopedia opens converted encyclopedia, the stored procedures will be properly created with an owner of 'dbo'.

Upgrading to SA V10.4 and later – Converting SA-DOORS Link Info to SA-DOORS V2

**Conversion
Procedure to
Upgrade
Encyclopedias to
SA V10.4 and later**

If you have been using the SA-DOORS interface to link
model artifacts with DOORS objects, you will need to convert
your encyclopedia for use with the SA-DOORS V2 interface,
introduced with System Architect V10.4. To do so, perform
these steps:

Note: SAEM cannot perform standard functions on an
encyclopedia residing on a remote server.

1. We advise that you backup the encyclopedia used
   with the SA-DOORS V1 interface. You may backup
   the encyclopedia using SAEM (run SAEM, connect to
   your server, choose the encyclopedia, and select
   Database, Backup).

2. Run System Architect and open the encyclopedia
   that you wish to convert.

3. Run DOORS and open the database that the
   encyclopedia is associated with.

4. In DOORS, from the top menu select **SA,
   Upgrade/Recover System Architect**.

**SA-DOORS
Interface --
Updating DOORS-
linked
Encyclopedias for
Use with SA V10.4
and later**

If you have a SA V10.3 encyclopedia that you linked to
DOORS, you will need to update that encyclopedia with
SAEM to reestablish the link. To do so, perform these steps:

1. Backup the encyclopedia (see above).

2. Launch SAEM (SQL Server) or SAEM (Oracle),
   depending on the your server.

1-11

3. Click Tools menu, and select Update DOORS. The message pane at the bottom of the SAEM window confirms your results.

4. Open the encyclopedia with System Architect V10.4 and later. The links should now be updated to work with SA V10.4 and later.

Adjusting Macros that Run Off Menus in SA V10.1

Conversion Procedure for Macros Associated with Menu Items in SA V10.1

As of V10.1 System Architect has almost completely released the menus and toolbars from their previous fixed format. This means that users are now able to customize menus, and their customizations will remain. The only exceptions to this are the Draw menu and toolbar, where customization is only possible after the drawing tool items.

The goal of this change was to enable more tailored menu and toolbar configurations for specific user groups via SA Catalog Manager.

In general users should notice no significant changes to menus in this release of System Architect. All menu items and toolbars will remain where they were in previous versions.

If you have created custom macros that run off menu items, you will need to adjust them slightly for use in SA V10.1 and later. Details are provided in Chapter 3 of this manual.

Opening V9 Encyclopedias with System Architect V10

Procedure Required for Opening V9 Encyclopedias with V10

System Architect V10 requires a GetFileSize stored procedure in all encyclopedias (SQL Server 2000 or MSDE databases) which System Architect V9 did not require. System Architect will automatically try to add this stored procedure when the V9 encyclopedia is initially opened with V10. We recommend that the first person to open the V9 encyclopedia with V10 has a System Administrators role, or is the owner of the encyclopedia (you may find out who the owner of an encyclopedia is using SAEM (for MSDE) or Microsoft's Enterprise Manager (for SQL Server). When either of these types of users opens a V9 encyclopedia with V10, the GetFileSize stored procedure will be properly created with an owner of 'dbo'.

After the stored procedure is created (with an owner of 'dbo'), the System Architect administrator must also grant Execute privileges on the GetFileSize stored procedure to all users -- whether that is accomplished by granting it to their group, role, or individual account, depending on how they have set up security on SQL Server.

If a user other than the System Administrator or the owner opens the encyclopedia initially, executive rights to the stored procedure will be granted to that user only, and no other users will be able to access the encyclopedia. The reason is that if the GetFileSize stored procedure is created with an owner other than 'dbo', then System Architect will not recognize the stored procedure. When other users open the encyclopedia, they will get an error message indicating that the GetFileSize stored procedure wasn't found.

Conversion from V8 to V9, V10 or later – From a dBase to a SQL Server Encyclopedia

Procedure Required for Opening V8 Encyclopedias with V9, V10, or Later

System Architect versions V9, V10, and later offers a new underlying repository structure from V8 and earlier of the product. Starting with V9, encyclopedias are built as databases on SQL Server. You may create encyclopedias on either SQL Server 2000 or Microsoft Server Desktop Engine (MSDE). The procedure to convert existing V8 encyclopedias to V9, V10, and later is provided in Chapter 2.

Convert By Using Encyclopedia Merge Facility

In general, you create a new encyclopedia on SQL Server 2000 (or MSDE), and merge the older V8 dBase encyclopedia contents into it.

Changes to IMF Calls Affect VBA Macros

If you have created custom VBA macros for System Architect, and use certain IMF calls, you will need to make some minor modifications to those macros before proceeding. These changes are also detailed in Chapter 2.

Conversion to SA V8 – GUIDs and UML

Procedure Required for Opening SA 2001 Encyclopedias with V8

The conversion routine to bring encyclopedias created with System Architect v6 or v7 to V8 status has two broad functions to perform:

- assign a GUID to every object,

- convert any existing UML model elements to the new.

GUIDs

Every diagram, symbol and definition in the encyclopedia being converted will be given a Global Unique Identifier or GUID. Note that this applies to **all** objects, whether they are part of UML or otherwise. The GUID is a 64 character long value provided by Microsoft code. The value is supposed to be unique throughout the known universe and for all time.

UML

System Architect V8 features a major upgrade to UML functionality in the product. The UML upgrade features representational consistency of relationships drawn between UML classifiers, a new package structure (with many UML classifiers now 'keyed' to the package they are contained in), and new handling of implementation language types. Because of these and other features, a conversion of UML diagrams and definitions is necessary. Detailed information on the UML Conversion is provided in Chapter 5.

Conversion to SA/2001 – Data Modeling and Long File Names

In general there are two areas of change in System Architect that requires conversion for work done in version 4.0 of the product (or SA/BPR 3.2):

- Long Names (up to 80 characters) are now allowed for Symbols, Definitions, and Diagrams (the previous limit was 31 characters), and

- Data modeling has changed significantly in SA/2001.

When conversion is run against an encyclopedia created with a previous version of System Architect, a separate conversion routine is run for the each of the two topics above.

Long File Names

System Architect supports names up to a maximum of 80 characters long, compared to the old limit of 31 characters. The conversion routine converts all underlying fields so that they can hold the additional characters. Existing names of symbols will not change. You will, after the conversion process is complete, be able to rename old names to new names that are up to 80 characters long.

Data Modeling

Data modeling functionality was greatly enhanced in System Architect 2001 and later. Data models built with previous versions of System Architect (4.0 and 3.1, etc) must be run through a conversion routine to make them compatible with data modeling in System Architect. Details about the changes to data modeling functionality and the specifics of what the conversion program does are provided in chapters 7 and 8 of this manual. An overview is provided here.

Special Note for SA/Catalyst Users

In addition, changes have been made to some of the names of business modeling diagrams and definitions for the Catalyst methodology. This only affects users of SA/Catalyst who have made changes to the properties file (USRPROPS.TXT). Users who have modified properties for SA/Catalyst should read the conversion instructions in

1-17

chapter 2, and refer to the name changes in chapter 9 of this manual to see if the name changes affect them.

Special Note for SA/BPR Users

In addition SA/BPR users should be aware that IDEF1X data models are not directly converted to the new data modeling format. They can be converted if they are first mapped to physical diagrams in the SA/BPR tool, converted to physical diagrams in SA/2001, then mapped to ER diagrams in SA/2001. In addition, Swimlanes from IDEF and Organizational Units from the Catalyst methodology are both present in SA/2001. SA/BPR users who have modified the properties of a Swimlane definition through USRPROPS.TXT should pay special attention to the conversion instructions for SA/BPR users in Chapter 2.

2

Procedures for Converting Existing Encyclopedias

Introduction

This chapter provides instructions on how to convert encyclopedias created with previous versions of products in the Rational System Architect suite to encyclopedias for use with Rational System Architect V8.

Overview

Rational System Architect V10.3 creates new stored procedures and tables (for history) and makes changes to existing stored procedures in all encyclopedias. Because of this, encyclopedias created in V10.1, V10.0, or V9 of the product must be converted/upgraded for use in Rational System Architect V10.3. The instructions on how to do this follow in the next section.

Starting with Rational System Architect V9, a new underlying repository structure was introduced into the tool – encyclopedias are built as databases on SQL Server. You may create encyclopedias on either SQL Server 2000 or Microsoft Server Desktop Engine (MSDE). Encyclopedias used with Rational System Architect V8 were based on a proprietary format based on dBase. They must be converted to SQL Server for use with Rational System Architect V9, V10 and later.

Opening a V9 Encyclopedia with V10

Rational System Architect V10 requires a GetFileSize stored procedure in all encyclopedias (SQL Server 2000 or MSDE databases) which Rational System Architect V9 did not require. Rational System Architect will automatically try to add this stored procedure when the V9 encyclopedia is initially opened with V10. After the stored procedure is created (with an owner of 'dbo'), the Rational System Architect administrator must also grant Execute privileges on the GetFileSize stored procedure to all users. More details are provided in Chapter 1 of this manual.

Converting V8 to V9/V10 and Later

To convert an encyclopedia for use with V9, V10, and later versions of Rational System Architect, it must be in V8 format first. Instructions to do this are provided in the next section.

Converting Previous Versions to V8

Rational System Architect V8 has an automatic conversion utility that converts encyclopedias created in previous versions of the product. However, because previous versions of the product enabled the user to model inconsistent information in some areas, the conversion utility may come across decision points during conversion that need human user input. The conversion will point out modeling

discrepancies -- it is then up to the user to fix these discrepancies.

Maintaining an Old Version of SA 2001 Is Recommended

The recommended place to fix a discrepancy is in the original encyclopedia. Because of this, it is recommended that a version of Rational System Architect 2001 be maintained by the user if possible, on a separate machine. Conversion can then be run against the encyclopedia using Rational System Architect V8, and if discrepancies are found, the encyclopedia can be moved to the machine with Rational System Architect 2001, the discrepancies can be fixed, and the encyclopedia can then undergo a final conversion with Rational System Architect V8.

Amalgam of Two Conversions -- Long File Names & Data Modeling; UML & GUIDs

The conversion routine accessed is accessed in Rational System Architect by selecting **Tools, Conversion** (with any encyclopedia open). The conversion routine is actually an amalgam of two conversion routines – a data-modeling/long-names conversion to convert encyclopedias created with versions of Rational System Architect prior to SA/2001 (ie, SA 4.0) to Rational System Architect 2001 status, and a GUID/UML conversion to convert encyclopedias created with Rational System Architect 2001, v6 or v7, to Rational System Architect V8.

When you select **Tools, Conversion**, the conversion routine will ask you to specify the encyclopedia to be converted, examine the encyclopedia, and determine what type of conversion is necessary. If the encyclopedia was created with a version of Rational System Architect prior to SA/2001, the data-modeling/long-names conversion will run first, to convert the encyclopedia to SA/2001 status. When that conversion finishes, the GUID/UML conversion will run to convert the encyclopedia to SA/2001 V8 status. If the encyclopedia was created with SA/2001 v6 or v7, only the GUID/UML conversion will run to bring it to V8 status.

The procedure for the data-modeling/long-names conversion routine is essentially the same no matter what encyclopedia you convert. However, there are nuances involved with each encyclopedia created by a different Rational System Architect product.

This chapter is divided into four sections, the first detailing the procedure for converting SA/2001 v6 or v7 encyclopedias to

2-3

V8 status, and three other sections, each one giving the specific procedure for performing a data-modeling/long-names conversion for a particular Rational System Architect product. There is a section for converting encyclopedias created with Rational System Architect 4.0 products (including those created with SA/Object Architect 4.0, SA/Data Architect 4.0, and SA/Catalyst 4.0). There is a section for converting encyclopedias created with SA/BPR 3.2. There is also a section for converting encyclopedias created with SA/Data Architect 4.5.

Upgrading to SA V10.3 – Converting SA V10.1, V10.0, and V9 Encyclopedias

Conversion Procedure to Upgrade Encyclopedias to SA V10.3

Rational System Architect V10.3 creates new stored procedures and tables (for history) and makes changes to existing stored procedures in all encyclopedias (SQL Server 2000 or MSDE databases). Because of this, encyclopedias created with V10.1, V10.0, or V9 of the product must be converted to V10.3 format. This is a two-stage process:

Stage 1 – convert the data of the encyclopedia using SAEM, as follows:

1. Run Rational System Architect Encyclopedia Manager (SAEM) (select **Start, Programs, IBM Rational, IBM Rational Lifecycle Solutions Tools, IBM Rational System Architect 11.3.1, SAEM**).

2. In SAEM, log in to the server containing the encyclopedia you wish to upgrade (select **Server, Connect**).

3. Choose the encyclopedia you wish to upgrade (select **Database, Select Database**).

4. In SAEM, select **Tools, Convert to 10.3 Format**. This will convert the data to the format necessary to be used with Rational System Architect 10.3.

Stage 2 – A person with a System Administrator's role, or the owner of the encyclopedia, must open the converted encyclopedia in Rational System Architect V10.3, as follows:

1. Find out who the owner of an encyclopedia is using SAEM (for MSDE) or Microsoft's Enterprise Manager (for SQL Server). Have that person or a person with a System Administrator's role, open the converted encyclopedia in SA V10.3.

2-5

2. Rational System Architect will automatically try to add new stored procedures and tables to the existing encyclopedia when it is initially opened with SA V10.3. As long as a person with a Systems Administrator role or the owner of the encyclopedia opens converted encyclopedia, the stored procedures will be properly created with an owner of 'dbo'.

Upgrading to SA V10.3 – Converting SA-DOORS Link Info to SA-DOORS V2

Conversion Procedure to Upgrade Encyclopedias to SA V10.3

If you have been using the SA-DOORS interface to link model artifacts with DOORS objects, you will need to convert your encyclopedia for use with the SA-DOORS V2 interface, introduced with Rational System Architect V10.3. To do so, perform these steps:

1. We advise that you backup the encyclopedia used with the SA-DOORS V1 interface. You may backup the encyclopedia using SAEM (run SAEM, connect to your server, choose the encyclopedia, and select **Database, Backup**).

2. Run Rational System Architect and open the encyclopedia that you wish to convert.

3. Run DOORS and open the database that the encyclopedia is associated with.

4. In DOORS, from the top menu select **SA, Upgrade/Recover System Architect**.

2-7

Opening V9 Encyclopedias with Rational System Architect V10

Procedure Required for Opening V9 Encyclopedias with V10

Rational System Architect V10 requires a GetFileSize stored procedure in all encyclopedias (SQL Server 2000 or MSDE databases) which Rational System Architect V9 did not require. Rational System Architect will automatically try to add this stored procedure when the V9 encyclopedia is initially opened with V10. We recommend that the first person to open the V9 encyclopedia with V10 has a System Administrators role, or is the owner of the encyclopedia (you may find out who the owner of an encyclopedia is using SAEM (for MSDE) or Microsoft's Enterprise Manager (for SQL Server). When either of these types of users opens a V9 encyclopedia with V10, the GetFileSize stored procedure will be properly created with an owner of 'dbo'.

After the stored procedure is created (with an owner of 'dbo'), the Rational System Architect administrator must also grant Execute privileges on the GetFileSize stored procedure to all users -- whether that is accomplished by granting it to their group, role, or individual account, depending on how they have set up security on SQL Server.

If a user other than the System Administrator or the owner opens the encyclopedia initially, executive rights to the stored procedure will be granted to that user only, and no other users will be able to access the encyclopedia. The reason is that if the GetFileSize stored procedure is created with an owner other than 'dbo', then Rational System Architect will not recognize the stored procedure. When other users open the encyclopedia, they will get an error message indicating that the GetFileSize stored procedure wasn't found.

2-8

Converting V8 to V9/V10

Encyclopedias created with versions of Rational System Architect prior to V9 were based on a proprietary format based on dBase. That format was held in a folder with various files inside of it. Beginning with V9, encyclopedias are now stored in databases that can reside on SQL Server, MSDE or Oracle.

Preparatory Step if You Have Custom Graphics

The conversion process is as follows:

1. If you have custom .bmp and .wmf files associated with an encyclopedia, you must perform two preparatory steps:

    a. Use Windows Explorer or equivalent, to copy the custom .bmp and .wmf files associated with the V8 encyclopedia to be converted to the images folder of the main Rational System Architect software directory (<C>:\Program Files\IBM\Rational\\System Architect Suite\11.3.1\System Architect\Images).

    b. Copy the usrprops.txt file of the V8 encyclopedia that contains custom images, and paste it into the main Rational System Architect directory (<C>:\Program Files\IBM\Rational\\System Architect Suite\11.3.1\System Architect), replacing the usrprops.txt file that is there.

Convert Encyclopedias By Using Merge Facility

1. In Rational System Architect, create a new encyclopedia in SQL Server or MSDE. (This new encyclopedia will now already contain all necessary custom pictures and usrprops.txt.). Procedures for creating a new encyclopedia in SQL Server or MSDE are provided in the on-line help or in the Installation Guide.

2. Use Rational System Architect's Encyclopedia Merge utility to merge all information from the V8 (dBase)

2-9

encyclopedia into the new V9/10 encyclopedia, as follows:

a. Open the encyclopedia you created in the above step.

b. Click **Tools**, and select **Merge**.

c. In the **Merge/Extract** dialog, click the **Browse** button. The **Open Encyclopedia** dialog will open.

d. Click on the **Existing** icon.

e. Click the '**…**' ellipses button to create a new connection. The **Connection Manager** dialog is displayed.

f. Click on the **New** icon. A new entry for your connection is created in the **Available connections** field.

g. Enter a name in the **Connection Name** column.

h. Click on the **Server Type** column and select **dBase**.

i. Click once on the **Server Name** column then click the '**…**' ellipses button that appears in that column.

j. Browse to the folder with your dBase encyclopedia, select it, and click **OK**.

k. Click **OK** to close the **Connection Manager**.

l. Select your dBase encyclopedia from the **Open Encyclopedia** dialog and click **OK**.

m. Select your merge options in the **Merge/Extract** dialog, and click **OK**. A reports window displays the results of the merge/extract operation.

Modification of VBA Macros and SA/Word Macros

Users who have built their own VBA macros for use with Rational System Architect, or their own SA/Word VBA macros (which use the sawd2001.dll file), need to be aware

2-10

of certain changes that have been made to Rational System Architect's IMF calls, and may need to modify their code appropriately. Please refer to Chapter 3 for details.

3

Modifying Macros for Use with V10.1

Introduction

This chapter provides details on converting V8 encyclopedias (dBASE) to V9 (SQL Server) and later. It also describes the workflow changes that users who have been using Rational System Architect V8 and before will encounter when using the V9 or later version of the product.

Topics in this chapter	Page
Modifying Macros that Run Off Menu System for V10.1	3-2

Modifying Macros that Run Off Menu System for V10.1

As of V10.1 Rational System Architect has almost completely released the menus and toolbars from their previous fixed format. This means that users are now able to customize menus, and their customizations will remain. The only exceptions to this are the Draw menu and toolbar, where customization is only possible after the default list of drawing tool commands or buttons.

The goal of this change was to enable tailored menu and toolbar configurations for specific user groups via SA Catalog Manager.

How Will this Affect Current Users?

In general users should notice no significant changes to menus in this release of Rational System Architect. All menu items and toolbars will remain where they were in previous versions, it is expected that the current layout will eventually be known as Classic Rational System Architect.

Users will however be able to customize their menus toolbars (except the Draw toolbar) and have menu items remain where they are place, without the need to add macro code.

How Will this Affect Macros Which Change Menus?

IBM's aim to ensure backward compatibility is always a top priority, however some improvements are not without side effects. The updates to the menu system have been quite extensive and have involved a change in philosophy, where items are no longer removed from the menu system. Instead, where a menu item would previously have disappeared, it is simply hidden. This change of emphasis does impact on code written to support macro menus items.

Previously macro menu code tended to be written taking into account the fact that Rational System Architect would destroy certain menus, and force them to be redrawn. This destruction meant that users who required menu items to disappear during a session, could save themselves the need to actually remove an item from a menu programmatically. A typical example might be when a diagram was closed – the tools menu would be destroyed and re-created. Thus the user would only be required to re-insert the menu item if it was required at that time. Now that Rational System Architect does not destroy menus, these menu items will remain visible.

Essentially all macro menu code which requires menu items to be removed from the menu system, will need code added to the menu event handler logic, to remove (hide) a menu item that should no longer be visible.

What Will Happen if No Change is Made to Macro Code Which Affects Menu Items ?

If you do not make any changes to macros that you have written to appear on menus, the following will be the consequences:

- If all the macro menu items are expected to remain available for the full session – the code will run as before, no change is required.
- If some macro items are expected to disappear when the menu they are on is 'redrawn' – these macro menu item will be shown the first time they are activated and remain visible for the duration of the session.

How Can I Change the Macro to Follow the New Regime?

The changes are quite simple to implement.

1. Identify the piece of code which inserts the item in the menu
2. Add an Else clause, which is operated when the macro should not be shown.
3. In the Else clause add code to remove the item from the menu. This will hide the item.

Example

Here is an example:

```
Option Explicit
Public WithEvents App As Application  ' The application which will raise events
Private Sub App_ReportsMenuUpdate()
  ' This event happens whenever a diagram is shown or hidden.
    UpdateSamplePopupMenu
End Sub
Public Sub UpdateSamplePopupMenu()
    Dim lngErr As Long
    If Not Application.Encyclopedia Is Nothing Then
      If Not Application.Encyclopedia.GetCurrentDiagram() Is Nothing Then
        ' Insert our popup menu 'MENUSAMPLEPOPUP' in the 'Reports' menu,
before the 'Print all Diagrams' item.
        lngErr = Application.InsertPopupMenuItemInMenu(MENUSAMPLEPOPUP,
"&Reports", "Print all &Diagrams")
      Else
        ' Remove our popup menu 'MENUSAMPLEPOPUP' from the menu system
        lngErr = Application.RemoveItemFromMenu(MENUSAMPLEPOPUP, "")
      End If
    End If
End Sub
```

The code in **bold** above has been added to ensure the macro item is removed (hidden) from the menu system, where previously it would have been removed by Rational System Architect's menu redrawing technique. The use of the menu name to remove from is not required as it is no longer used.

3-3

How do I Reset My Toolbar to the Default State ?

There are two ways to reset the toolbar system.

- Complete reset
  - o This removes all customizations from the menu system and returns the menu system to initial settings.
- Macro item reset
  - o This removes all macro menu customization and tools, but retains any product menu customization.

To perform a complete reset:

1. Right click on the toolbar
2. Select Customize
3. Select the Toolbars tab
4. Select any toolbar from the list
5. Press the 'Reset' button
6. You will be given a warning message. Respond with Yes.
7. Close the Customize dialog
8. Restart SA

To perform a macro menu reset:

1. From the 'Tools' menu select the 'Macros' popup
2. From the 'Macros' popup select the 'Macro Projects' dialog
   a. *If a particular macro project is causing problems, uncheck it's active state in the dialog, and press Apply.*
3. Choose the icon at the top left of the dialog with the X through a menu – it has the tooltip 'Reset macro menu customizations'
4. You will be given a warning message. Respond with Yes.
   a. *If a particular macro project was causing problems, recheck it's active state in the dialog, and press Apply.*
5. Close the Macro Projects dialog
6. Restart SA (not essential, but some macro projects may not reload their menus until after a restart)

3-4

What are the New Guidelines for Adding Macro Items to Menus Programmatically?

How Do I Add a Macro to a Standard Rational System Architect Menu ?

The main SA2001.Application method to support the adding of macro items is:

InsertMacroItemInMenu(MacroName as String, MacroItemCaption as String, InMenuTitleCaption as String, [BeforeMenuItemCaption as string]) as long

This function adds the MacroName (which comprises - "<Project>.<Module>.<Subroutine>") using the caption MacroItemCaption to the SA/User menu titled InMenuTitleCaption, before the menu item titled BeforeMenuItemCaption. If no Before item is specified then the item is added to the end of the menu. If the before item is #TOP# then the item is inserted at the top of the menu.

The function returns 0 if successful, otherwise none zero.

In other words:

Set App = New SA2001.Application

x = App.InsertMacroItemInMenu("MyProject.MyModule.MySub", "&Test Item", "&Tools","&Macros")

This inserts the item 'Test Item' in the Tools menu, before the Macros popup item. When the item is clicked on, the subroutine 'MySub' is executed in the module 'MyModule' in the project 'MyProject'.

For this to succeed the subroutine specified MUST already exist. Also, the captions MUST be specified exactly as in the menu.

For example,Tools is represented as the string &Tools

An ampersand implies an underscore under the following letter. This IS case sensitive.

The subroutine name, should NOT have ()'s following it.

The adding of BMPs to menu items is done by the AssignBMPtoMacroItem method.

AssignBMPtoMacroItem(MacroName as String, BMPFileName as String) as long

This function returns zero if successful, or none zero otherwise.

This should be done prior to adding the macro item to the menu - but need only be done once.

3-5

For example:

Set App = New SA2001.Application

x = App.AssignBMPtoMacroItem("MyProject.MyModule.MySub", "C:\Piccy.BMP")

x = App.InsertMacroItemInMenu("MyProject.MyModule.MySub", "&Test Item", "&Tools","&Macros")

This will add the piccy.bmp to the MySub macro, then add it to the menus as in the previous example. Note that the customize menus (right click on menus) option will now include the Piccy.BMP and the caption '&Test Item', if you wished to put the menu item anywhere else in your menus.

Using the Events

The following SA2001.Application events may assist macro code when deciding when to make menu visibility changes:

- MainMenuUpdate: When the whole menu is affected

- MethodMenuUpdate: When dictionary type menu items are updated

- ToolsMenuUpdate: When tools type menu items are updated

- ReportsMenuUpdate: When report type menu items are updated

- App_ShutDown: When SA shuts down

Events are NO LONGER required to keep user items in the menus. However if you wish to remove items from a menu, the following sections of code provide the outline for such event handling. (Note: The Project is called 'MyProject')

MODULE - AutoExec

```
' Main module which maintains the VBA based event handler
Dim EventHandler As EventCls
' Main Subroutine which starts the event handler
Sub Main()
  Set EventHandler = New EventCls     ' Create an event handler
  Set EventHandler.App = Application  ' Connect the event handler to SA
  ' Perform any additional start up work here
  ' This is a good place to add you BMPs and create your PopUp menus
  InitBMPs
  InitPopUps
End Sub

Sub InitBMPs()
  Dim x As Integer
  x =
Application.AssignBMPtoMacroItem("MyProject.MyModule.PRINTDIA
GRAMS", "SAWORD.BMP")
End Sub

Sub InitPopUps()
  Dim x As Integer
  ' NB: Once a popup is created, it remains until it is removed by
'RemovePopupMenu', and can be added/removed from menus at will.

  ' Create the popup menu with its bitmap
  x = Application.CreatePopUpMenu("Sample
Macros","SAWORD.BMP")
  ' Add the item to the popup
  x =
Application.InsertMacroItemInMenu("MyProject.MyModule.PRINTDIA
GRAMS", "&Print a Diagram", "Sample Macros")
End Sub
```

CLASS - EventCls

```
Public WithEvents App As Application  ' The application which will
raise events

Private Sub Class_Initialize()
  ' No need to do anything
End Sub

Private Sub Class_Terminate()
  ' No need to do anything
End Sub
```

```
Private Sub App_ReportsMenuUpdate()
 ' Reports menu has been redrawn
 Dim x As Long
 x = App.SetSeparatorBefore("&Report Generator...", "&Reports",
True)

 ' Insert a menu item in the 'Reports' menu, before the 'Report
Generator...' item, called 'Print all Diagrams' which calls
PrintDiagrams2
 x =
App.InsertMacroItemInMenu("MyProject.MyModule.PRINTDIAGRAMS
2", "Print all &Diagrams", "&Reports", "&Report Generator...")

 ' Insert our popup menu 'Sample Macros' in the 'Reports' menu,
before the 'Print all Diagrams' item.
 ' NB: You should not keep destroying and creating popups here,
unless you have a good reason - do this once at initialization.
 x = App.InsertPopupMenuItemInMenu("Sample Macros", "&Reports",
"Print all &Diagrams")
End Sub
```

MODULE - MyModule

```
Public Sub PrintDiagrams()
' User Code to run
MsgBox "Print Diagrams"
End Sub
Public Sub PrintDiagrams2()
' User Code to run
MsgBox "Print Diagrams2"
End Sub
```

How Do I Add a Popup ?

Use the CreatePopUpMenu method with the name you wish to use for your popup. And a BMP if desired. The popup is added to the root collection of popups in SA.

When you leave SA you no longer need your event handler class to remove any added popups using RemovePopUpMenu

How Do I Add Macro Items to a Popup ?

In exactly the same way as you do for adding to SA menus.

How do I Add a Popup to an SA Menu ?

You may add a popup to a Rational System Architect menu in exactly the same way you add a macro item to a menu, except, use InsertPopupMenuItemInMenu and do not specify a macro name.

For example:

InsertPopupMenuItemInMenu(PopUpName as String, InMenuTitleCaption as String, [BeforeMenuItemCaption as string]) as long

How Do I Add a Separator to a Menu Item ?

Use SetSeparatorBefore, provide the menu name and item name, then set True for a separator and false for no separator.

How Do I Remove an Item from a Menu ?

You can remove popups or menu items using the same method: use RemoveItemFromMenu specify the name of the menu and the name of the item.

Note: You can only remove items that you have added. For example, Rational System Architect menu items cannot be removed. The items are not actually removed, they are instead hidden. This allows users to customize the items to be elsewhere in the menu system. Note you can specify "" for the Menu name, as the Menu name is now ignored. The item will be hidden wherever it is in the menu system.

What Does RemovePopUpMenu Do ?

Since all popups and tools which are added to the menu system will remain even after SA has been restarted, it may become necessary to remove a popup menu which is no longer required. The App.RemovePopUpMenu(<PopupName>), method will totally remove the popup from the menu system.

If this method is used in a macro, say at SA shutdown, the popup and all its customization will be removed everytime SA shuts down. This would have the effect, that customization of this popup is not kept between SA sessions.

4

Rational System Architect V9, V10, and Later Conversion

Introduction

This chapter provides details on converting V8 encyclopedias (dBASE) to V9 (SQL Server) and later. It also describes the workflow changes that users who have been using Rational System Architect V8 and before will encounter when using the V9 or later version of the product.

Converting Encyclopedias

Encyclopedias created with versions of Rational System Architect prior to V9 (which were in a proprietary format based on dBase) need to be converted to SQL Server or MSDE. We repeat the steps presented in Chapter 2, on how to do this. Later in this chapter, details are provided on changes that may be necessary to user-created VBA macro files.

Preparatory Step if You Have Custom Graphics

The conversion process is as follows:

1. If you have custom .bmp and .wmf files associated with an encyclopedia, you must perform two preparatory steps:

    a. Use Windows Explorer or equivalent, to copy the custom .bmp and .wmf files associated with the V8 encyclopedia to be converted to the images folder of the main Rational System Architect software directory (<C>:\Program Files\IBM\Rational\System Architect Suite\System Architect\Images).

    b. Copy the usrprops.txt file of the V8 encyclopedia that contains custom images, and paste it into the main Rational System Architect directory (<C>:\Program Files\IBM\Rational\System Architect Suite\System Architect), replacing the usrprops.txt file that is there.

    These two steps provide a shortcut to making the SQL Server or MSDE encyclopedia that you create, ready for conversion

Convert Encyclopedias By Using Merge Facility

2. In Rational System Architect, create a new encyclopedia in SQL Server or MSDE. (This new encyclopedia will now already contain all necessary custom pictures and usrprops.txt.). Procedures for creating a new encyclopedia in SQL Server or MSDE are provided in the on-line help or in the Installation Guide.

3. Use Rational System Architect's Encyclopedia Merge utility to *merge* all information from the V8 (dBase) encyclopedia into the V9 encyclopedia, as follows:

    a. In Rational System Architect, select **Tools, Merge**.

b. In the **Merge/Extract** dialog, click the **Browse** button. The **Open Encyclopedia** dialog will open.

c. In the **Open Encyclopedia** dialog, toggle on **dBase Encyclopedia** in the **Existing** tab, and navigate to the path of the V8 encyclopedia to be converted. Select **Open**.

d. Back in the **Merge/Extract** dialog, click on the **Everything** button.

e. Keep the **If Items are identical** choice at its default "**Use the more recent**".

f. Click **OK** in the Merge/Extract dialog. The encyclopedia will be converted for use with Rational System Architect V9 – it will become a SQL Server database.

Modification of VBA Macros and SA/Word Macros

Users who have built their own VBA macros for use with Rational System Architect, or their own SA Word VBA macros (which use the sawd2001.dll file), need to be aware of certain changes that have been made to Rational System Architect's IMF calls, and may need to modify their code appropriately. Details are provided in the following sections.

Changes to VBA Macros

Changes necessary to VBA macros for use in Rational System Architect V9 can be summarized as follows:

- **Microsoft SQL Server Encyclopedia Access:** In version 9 of Rational System Architect the concept of a network lock file does not exist with respect to Microsoft SQL server encyclopedias. To ensure some form of exclusive use of the encyclopedia in order to avoid locking and collisions issues, two new functions were implemented: *CriticalRegion* and *AccessDefinition.*

- **ISAImf Objects:** Existing macros that use the *SA2001.ISAImf* object must have their Set statement changed to prevent a 'Type mismatch' run-time error.

- **SA2001.Encyclopedia.Fullname Property:** If a macro has used the *path* to the Encyclopedia *folder* to locate and access a file, such as 'D0000001.wmf' or 'Usrprops.txt', then it will need to be modified for use with V9 encyclopedias, which are maintained within SQL Server databases and store such files in a Files table. The required file will need to be *exported* before it can be used. The ISAImf object has an *SAFileExport* function that allows this.

- **ISAImf.SAGetEncyclopediaPath:** For the same reasons as those given for not using .Fullname (above), the *GetEncyclopediaPath* call cannot be used to locate and access encyclopedia files.

- **Importing Files:** In some cases it may be necessary to import files into the encyclopedia, so that an internal reference can be made. An example of this is when the .SetField() subroutine is used to link a picture symbol with its image file.

Details on the changes necessary for the situations described above are provided in the paragraphs that follow.

Microsoft SQL Server Encyclopedia Access

In version 9 of Rational System Architect the concept of a network lock file does not exist with respect to Microsoft SQL server encyclopedias.

To ensure some form of exclusive use of the encyclopedia in order to avoid locking and collisions issues, two new functions were implemented: **CriticalRegion** and **AccessDefinition**.

The following sequences of code in VBA macros should be replaced, wherever possible, by calls to AccessDefinition (described below) with the appropriate directive flags.

- FindDefinition, CreateDefinition

- SAFindDefinition, CreateDefinition, OpenDefinition

- FindDefinition, OpenDefinition

- FindDefinition, OpenLockDefinition

- CreateDefinition (where FindDefinition is not preceding it)

Any block of code relying on the fact that it had exclusive access to the encyclopedia (by locking it) and has modified records without locking them first, **must** be altered so that a lock is placed on the record before modifying it (OpenLockDefinition) and a call to CloseDefinition is present.

If after replacing the above types there are still sequences remaining, they should be bounded by a Critical Region.

We recommend that a full analysis of the VBA code be done before any changes are made.

1. CriticalRegion:

The **CriticalRegion** function has been implemented in order to avoid collisions on a SQL Server encyclopedia in a network environment. The function offers a form of exclusive access to the encyclopedia.

A critical region is an exclusive lock taken on a system resource so that any other process that attempts to start a critical region will wait until the current one is completed, thus allowing content sensitive processes to be single threaded. The call to end a Critical Region should only be made if the call to begin the Critical Region succeeded.begins or ends a critical region.

4-5

**int CriticalRegion(
BOOL bLock,   /* Input: action to take – start or
stop */ WORD IOErrorId)/* Input: Id for error
reporting*/**

Parameters:

bLock

Boolean indicating whether to stop (FALSE) or to start
(TRUE) the critical region

IOErrorId

Number ranging between 8600 and 9000 used for IO error
reporting.

Return Values:

If the function succeeds the return value is SUCCESS,
otherwise the value is DB_ERROR.

2. AccessDefinition:

The **AccessDefinition** function finds or creates, opens, or
open-locks a definition.

int AccessDefinition(

LPSAIDENT ddPrevDef,/* Input: previous definition */

LPCTSTR lpName,/* Input: definition name */

LPWORD lpwMinorType,/* Input: definition type */

**DWORD dwCommand,/* Input: combination of
commands */**

LPSAIDENT lpidDefinitionRet,/* Output: accessed
definition id */

**LPSAHANDLE lphDefHandleRet,/* Output:
accessed definition handle */**

**LPWORD lpwIsDefNew)/* Output: indicates the
creation of the definition */**

Parameters:

ddPrevDef

Pointer to the previous definition's ddId.
It can be NULL.

lpName

Name of definition to be opened. It can
be a qualified name.

LpwMinorType

Pointer to the minor type of the
definition to be opened.

dwCommand

Specifies the action to take. The following values can be
combined by using the bitwise OR operator:

Value	Meaning
IMF_DEFACCESS_CREATE	If definition attempting to be accessed couldn't be found, create it.
IMF_DEFACCESS_OPEN	Open definition.
IMF_DEFACCESS_LOCK	Lock the definition being opened. Used in conjunction with IMF_DEFACCESS_OPEN.

lpidDefinitionRet

Pointer to the opened/created definition's
ddId.

lphDefHandleRet

Pointer to the opened/created definition's
handle.

lpwIsDefNew

Pointer to a variable indicating whether or not we
created this definition.

Return Values

If the function succeeds, the return value is IM2XE_OK.

If the function fails, the return value is IM2XE_*. For information on the returned error, consult IM2XDEF.H located in the SADESIGN directory.

Remarks

The command to find the definition is implied. In other words this function can be invoked with a *dwCommand* of 0 to find a definition.

The AccessDefinition function sets its own critical region and was designed to replace sequences of the following types:

- FindDefinition, CreateDefinition

- SAFindDefinition, CreateDefinition, OpenDefinition

- FindDefinition, OpenDefinition

- FindDefinition, OpenLockDefinition

- CreateDefinition

When AccessDefinition is required to create a definition, it will actually write a record to the database. The function CreateDefinition did not (and still does not) do this; a call to SaveDefinition was needed to actually write the record to the database. Thus, any extant code sequence which used to call CreateDefinition (creates a memory definition block) and later decide whether to call SaveDefinition (writes out the record) or not, will have to be modified so that, if it is decided not to actually create this record, the code must now call DeleteDefinition to remove the unwanted record.

VBA macros that use IMF calls may require adjustment. The following IMF calls have been affected by the support for SQL Server databases:

- SAGetEncyclopediaPath

- Encyclopedia.Name

- Encyclopedia.Fullname

4-8

SAGetEncyclopediaPath: The problem with using this ISAImf call is the same as for Encyclopedia.FullName, instead of a folder reference, a file reference is supplied. This is due to the fact that an encyclopedia is now referenced through a .UDL file and not stored within a folder. This is likely to be used simply to provide information and should not present a problem.

Encyclopedia.Name: This has been difficult to trace – but normally used as information about the name of an Encyclopedia. The output will identify the UDL filename and not a subdirectory.

It is interesting to note that the ISAImf call SAGetEncyName returns the SQL Server's Database Name and not simply the UDL filename.

Encyclopedia.FullName: **As for SAGetEncyclopediaPath, a filename is returned and not a subdirectory name.**

ISAImf Objects

Existing macros that use the SA2001.ISAImf object must have their Set statement changed to prevent a 'Type mismatch' run-time error, as follows:

Previous Code:

If the existing macro had code as follows:

> Dim imfSA As SA2001.ISAImf
>
> ..

> **Set imfSA = SA2001.Application**

New Code:

It should be changed to the following for V9:

> Dim imfSA As SA2001.ISAImf
>
> ..

> **Set imfSA = appSA.Interface("ISAImf")**

SA2001.Encyclop edia.Fullname property

If a macro has used the *path* to the Encyclopedia *folder* to locate and access a file, such as 'D0000001.wmf' or 'Usrprops.txt', then it will need to be modified for use with V9 encyclopedias, which are maintained within SQL Server databases and store such files in a Files table.

4-9

The required file will need to be **exported** before it can be used. The ISAImf object has an **SAFileExport** function that allows this.

For example, if the code in the macro is as follows:

..

Let strEncyclopediaPath = appSA.Encyclopedia.FullName

Let strUSRProps = strEncyclopediaPath & "UsrProps.txt"

In Rational System Architect V9, the .Fullname property returns the UDL filename and its path. It might be preferable to use the **.Path** property that returns only the path to the UDL file. The path used to store the exported file could be any folder with write access.

The **SAFileExport** function takes the internal file reference and the external path and filename as parameters, returning 0 if the export is successful.

Let strPath = appSA.Encyclopedia.Path

Let strUSRProps = strPath & "USRProps.txt"

Let lngIMFerror = imfSA.SAFileExport("USRProps.txt", strUSRProps)

ISAImf.SAGetEncyclopediaPath

For the same reasons as those given for not using .Fullname (above), the **GetEncyclopediaPath** call cannot be used to locate and access encyclopedia files. The SAFileExport function should be used, as described above.

Importing Files

In some cases it may be necessary to import files into the encyclopedia, so that an internal reference can be made. An example of this is when the .SetField() subroutine is used to link a picture symbol with its image file, as follows

Call symPicture.SetField(SYMFLD_U_S1_ZPPICFILE, strFilePath)

The **SAFileImport** function takes the external path and filename and the internal file reference as parameters, returning 0 if the import is successful.

Let lngIMFerror = imfSA.SAFileImport(strFilePath, strInternalName)

..

Call symPicture.SetField
(SYMFLD_U_S1_ZPPICFILE, strInternalName)

Changes to SA Word Template Files

SAGetEncyclopediaPath   If an SA Word template uses the SAGetEncyclopediaPath function provided by sawd2001.dll file, to locate encyclopedia files, possibly for inclusion in the document, then these files need to be exported. The reason for this is that in V9, encyclopedias are maintained within SQL Server databases, not encyclopedia directory paths.

The **SAFileExport** function will allow the export of an internal file such as a diagram's .wmf file.

> Let intRet = SAGetEncyclopediaPath(strEncyclopediaPath, 255)
>
> Let strFilePath = strEncyclopediaPath & strFilename
>
> ..
>
> ActiveDocument.InlineShapes.AddPicture Anchor:=Selection.Range, _ FileName:= strFilePath, LinkToFile:= False, _ SaveWithDocument:=True

For Rational System Architect V9, it will be necessary to identify a suitable location for a temporary, exported file. Once the picture of the diagram has been inserted into the Word document the file will normally be deleted. The SAFileExport function takes the internal file reference and the external path and filename as parameters, returning 1 if the export is successful.

> Let strPath = "C:\temp\" ' or other suitable path
>
> Let strFilePath = strPath & strFilename
>
> Call SAFileExport(strFilename, strFilePath)
>
> ..
>
> ActiveDocument.InlineShapes.AddPicture Anchor:=Selection.Range, _
>
> FileName:= strFilePath, LinkToFile:=False, _
>
> SaveWithDocument:=True

Workflow Changes

This section describes workflow changes that users of Rational System Architect V8 and before will encounter when using Rational System Architect V9, V10, or later.

1. License Manager Must Run on Stand-Alone Machines with Rational System Architect

If you are running Rational System Architect on a stand-alone machine (such as a laptop), you will need to install and run License Manager on that machine before running Rational System Architect. Please see the Installation Guide.pdf for full instructions.

2. Creating and Opening Encyclopedias

Creating an Encyclopedia:

Creating an encyclopedia in Rational System Architect V9 differs from previous versions of the product. You either create an encyclopedia on an MSDE server, or a SQL Server 2000 server. Full details are specified in the on-line help and in Chapter 3 of the Installation Guide. This manual is provided in .pdf format on the installation Documentation DVD. If you do not have access to the Documentation CD, please contact your System Administrator.). A brief summary is provided here:

1. Select the **File, Encyclopedia Open** command.

2. In the **New** tab of the **Open a Project** dialog, specify a **Server Name**.

   For SQL Server 2000:

   Your System Administrator should provide you with this information -- which server is being used to create Rational System Architect encyclopedias.

   For MSDE:

   With MSDE installed on your local machine, the name of your machine should be represented in the **Server Name** drop-down list. You may either select the name of your machine, or **local** – both result in opening the MSDE server installed on your local machine. For MSDE is

installed on the network, you must select the appropriate server.

3. 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.

Important Note:

To create/open Rational System Architect encyclopedias on SQL Server 2000 in a network environment, the System Administrator **must set up appropriate access rights** on SQL Server 2000 for all Rational System Architect users. SA users will not be able to create encyclopedias unless they have proper rights to create databases on the server. They wont be able to open an encyclopedia unless they have rights to the database where the encyclopedia resides.

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

Database Access: For users to be able to open an encyclopedia and read and create definitions within it, they must have Database Access of db_datareader and db_datawriter. This is granted to the user by the System Administrator through SQL Server 2000 Enterprise Manager.

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:

   i. Toggle on Windows NT authentication, or

   ii. 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.

4. Specify the name of the database to create on that server. A Universal Data Link (UDL) file is created which contains the connection string to the server and database. The file can be created anywhere. Once created, the UDL file always correlates to a specific Rational System Architect encyclopedia on a specific server (so it makes sense to name it the same as the encyclopedia you are creating).

Opening an Encyclopedia:

Once created, opening an existing encyclopedia is performed via System Archtiect's Open a Project dialog (File, Open Encyclopedia), and selecting the .udl file of the encyclopedia. You may use browse button of the Existing tab in the Open a Project dialog to find and select the path and .udl filename of the encyclopedia, or use the Recent tab to select recent projects.

3. Encyclopedia Maintenance

Verify and Repair -- you may still run the Verify and Repair facility on an encyclopedia by selecting it from the Tools menu.

Dictionary Update -- you may still run the Dictionary Update facility on an encyclopedia by selecting it from the Tools menu.

Reorganize -- the Reorganize command is no longer available from the Tools menu. The Reorganize command used to reorganize and compact a dBase encyclopedia. This function is now performed by *shrinking* the database, either using SQL Server's Enterprise Manager tool, or IBM Rational's SAEM (Enterprise Manager) tool for MSDE. Instructions on how to shrink a database are provided in the on-line help of each respective tool.

Encyclopedia Comparisons Using SAEncomp -- SAEnComp, the encyclopedia compare facility, cannot currently work directly on encyclopedias created on SQL Server or MSDE in Rational System Architect V9. To compare encyclopedias using SAEnComp, you must extract those encyclopedias to a dBase version using Rational System Architect's Extract facility (Tools, Extract command). A **Working Document** on how to perform this is provided in the downloads area of the IBM IBM Rational website.

4. Customizing an Encyclopedia's Metamodel

Editing USRPROPS.TXT:

The User Properties (USRPROPS.TXT) file is used to modify and extend the metamodel of an encyclopedia. It now resides in the Files table of the encyclopedia database. To edit it, you must perform the following steps:

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 (you may change this setting via Tools, Session Options).

3. Edit the file as appropriate. (For more information about editing the USRPROPS.TXT file, refer to the Rational System Architect Extensibility Guide manual which is provided in .pdf format on the installation **Documentation DVD**. If you do not have access to the Documentation CD, please contact your System Administrator.)

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.

Images on Diagrams:

Images Depicting Symbols (USRPROPS.TXT Depiction Command):

User-defined images (WMF files to depict symbols on diagrams and BMP files for their respective representation on toolbars) are now housed in the Files table of an encyclopedia database. Such graphic files referenced in the USRPROPS.TXT of an encyclopedia must be imported into the Files table of the encyclopedia using the Encyclopedia File Manager command (Tools, Encyclopedia File Manager), or SQL Server 2000's Enterprise Manager program, or if you are using MSDE, the Rational System Architect Encyclopedia Manager utility (SAEM).

Picture File Copy:

Copying a picture from an external directory onto a Rational 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 Rational 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 Rational System Architect (Draw, Picture, File Reference). Rational 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 Rational System Architect diagram will disappear.

Encyclopedia File Manager:

There is a new Encyclopedia File Manager command in Rational System Architect, accessed via the Tools menu. This command enables you to import/export files to/from the **Files** table, **Error_Log** table, **Info_Log** table, and **Network_Log** table of an encyclopedia.

The Files Table:

The Files table contains the following files:

- **Graphics that You Paste or Copy Into an Encyclopedia:** A WMF or BMP file is created automatically every time you paste or file-copy a picture-type symbol into a diagram. One WMF or BMP is created for each picture. You can turn off the automatic creation of WMF files in SA2001.INI

- **Graphics that You Specify Via Depictions Clause in USRPROPS.TXT:** Rational System Architect enables you to specify new graphics for symbols drawn in an encyclopedia. You specify this using the Depictions clause in USRPROPS.TXT. You may specify new .BMP files for images on toolbars, and new .WMF files for use on diagrams. To make this happen, you must make the necessary changes to USRPROPS.TXT, and import the respective .BMP and .WMF files into the Files table of the encyclopedia.

- **The Property Set Files:** Three property set files determine the metamodel and the diagram and property sets that are turned on for an encyclopedia. They are as follows:

4-17

SAPROPS.CFG – created by IBM Specifies the metamodel of the encyclopedia.

USRPROPS.TXT – users add code to this file to specify metamodel extensions that they require.

SADECLAR.CFG – specifies the diagram types and property sets that are turned on for the encyclopedia. Is set by the Tools, Customize Method Support command in System Architect.

- **The AUTOEXEC.CSV File:** When you start a new project, the AUTOEXEC.CSV 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.

- **Format File (AUTOEXEC.STY, or 'style sheet'):** The format file, which determines the look of symbols that are drawn on diagrams. In particular, it refers to 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:Documents and Settings\<user profile name>\Local Settings\Application Data\Telelogic\System Architect, and then create a setting such as FORMATFILE = <C>:\Program Files\IBM\Rational\System Architect Suite\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 (ie, Payroll.sty).

    - You may name the format file AUTOEXEC.STY and import a copy into the Files table of all your project encyclopedias (using Rational 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 Error_Log Table:

The Error_Log table contains the error log of all errors reported

4-18

during operation of Rational 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 the encyclopedia tests and thier results. The file gets created in the encyclopedia being tested and can also be opened with any text editor.

The Network_Log Table:

The Network_Log table contains all messages reported regarding network use of Rational System Architect's multi-user version. 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.

5. Rational System Architect Encyclopedia Manager (SAEM)

IBM provides a utility program, Rational System Architect Encyclopedia Manager (SAEM) for users creating encyclopedias on MSDE. The SAEM program provides various functions to Rational System Architect users so that they may perform external tasks on their encyclopedias, such as shrinking, attaching, and deleting databases on a server. Users creating encyclopedias on SQL Server 2000 may use SQL Server's Enterprise Manager program to perform these functions.

The SAEM program is installed separately and comes with its own help file.

6. Version Control and PVCS

Version control of an encyclopedia to PVCS is now performed outside of Rational System Architect. Since an encyclopedia now exists as a database in SQL Server 2000, you use SQL Server 2000's capabilities to version a database to PVCS. A **Working Paper** on this topic is available on the Downloads page of www.IBM.com.

7. SABindery 6

SABind 6 is no longer supported in Rational System Architect's bindery. The functionality provided by SABind 6 is now done outside of Rational System

Architect, using SQL Server 2000 database access functionality. A **Working Paper** on this topic is available on the Downloads page of www.ibm.com.

5

IBM support

Introduction

There are a number of self-help information resources and tools to help you troubleshoot problems. If there is a problem with your product, you can:

Refer to the release information for your product for known issues, workarounds, and troubleshooting information.

Check if a download or fix is available to resolve your problem.

Search the available knowledge bases to see if the resolution to your problem is already documented.

If you still need help, contact IBM® Software Support and report your problem.

Contacting IBM Rational Software Support

If you cannot resolve a problem with the self-help resources, contact IBM® Rational® Software Support.

Note: If you are a heritage Telelogic customer, you can find a single reference site for all support resources at http://www.ibm.com/software/rational/support/telelogic/

Prerequisites   To submit a problem to IBM Rational Software Support, you must have an active Passport Advantage® software maintenance agreement. Passport Advantage is the IBM comprehensive software licensing and software maintenance (product upgrades and technical support) offering. You can enroll online in Passport Advantage at http://www.ibm.com/software/lotus/passportadvantage/howtoenroll.html

- To learn more about Passport Advantage, visit the Passport Advantage FAQs at http://www.ibm.com/software/lotus/passportadvantage/brochures_faqs_quickguides.html.

- For further assistance, contact your IBM representative.

To submit a problem online (from the IBM Web site) to IBM Rational Software Support:

- Register as a user on the IBM Rational Software Support Web site. For details about registering, go to http://www.ibm.com/software/support/.

- Be listed as an authorized caller in the service request tool.

Other information

For Rational software product news, events, and other information, visit the IBM Rational Software Web site: http://www.ibm.com/software/rational/.

Submitting problems

To submit a problem to IBM Rational Software Support:

1. Determine the business impact of the problem. When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem.

To determine the severity level, use the following table.

Severity	Description
1	The problem has a critical business impact: you are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
2	The problem has a significant business impact: the program is usable, but it is severely limited.
3	The problem has some business impact: the program is usable, but less significant features (not critical to operations) are unavailable.
4	The problem has minimal business impact: the problem causes little impact on operations or a reasonable circumvention to the problem was implemented.

2. Describe the problem and gather background information. When you describe the problem to IBM, be as specific as possible. Include all relevant background information so that IBM Rational Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?

- To determine the exact product name and version, use the option applicable to you:
    - Start the IBM Installation Manager and click **File** > **View Installed Packages**. Expand a package group and select a package to see the package name and version number.
    - Start your product, and click **Help** > **About** to see the offering name and version number.

- What is your operating system and version number (including any service packs or patches)?

- Do you have logs, traces, and messages that are related to the problem symptoms?

- Can you recreate the problem? If so, what steps do you perform to recreate the problem?

- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, or other system components?

3. Are you currently using a workaround for the problem? If so, be prepared to describe the workaround when you report the problem.

4. Submit the problem to IBM Rational Software Support in one of the following ways:

- Online: Go to the IBM Rational Software Support Web site at https://www.ibm.com/software/rational/support/. In the Rational support task navigator, click **Open Service Request**. Select the electronic problem

reporting tool, and open a Problem Management Record (PMR) to describe the problem.

- For more information about opening a service request, go to http://www.ibm.com/software/support/help.html.

- You can also open an online service request by using the IBM Support Assistant. For more information, go to http://www.ibm.com/software/support/isa/faq.html.
- By phone: For the phone number to call in your country or region, visit the IBM directory of worldwide contacts at http://www.ibm.com/planetwide/ and click the name of your country or geographic region.

- Through your IBM Representative: If you cannot access IBM Rational Software Support online or by phone, contact your IBM Representative. If necessary, your IBM Representative can open a service request for you. For complete contact information for each country, visit http://www.ibm.com/planetwide/.

6

Appendix:

Introduction

This chapter contains information about the legal uses and trademarks of IBM® Rational® System Architect®.

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or

implied warranties in certain transactions. Therefore, this statement may not apply to you.

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

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

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

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

Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, MA 02142
U.S.A

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been

estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Copyright license

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2000 2011.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.html

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

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