

Deployment of Enterprise Architect

by Dermot O'Bryan

All material (c) Sparx Systems 2005

<http://www.sparxsystems.com>

Table of Contents

INTRODUCTION	3
DEPLOYMENT FEATURES	3
CHOICE OF REPOSITORY	4
EAP FILE REPOSITORY	4
DBMS REPOSITORY	5
DEPLOYMENT SCHEMAS.....	5
SINGLE SITE SCENARIO	5
MULTIPLE SITE SCENARIOS	6
INTERCHANGING DATA BETWEEN .EAP FILES AND DBMS REPOSITORIES.....	8
APPENDIX 1: REPLICATION VS. XMI IMPORT/EXPORT	10
APPENDIX 2: PACKAGE CONTROL AND VERSION CONTROL	11
PACKAGE CONTROL	11
VERSION CONTROL	12
APPENDIX 4: ENTERPRISE ARCHITECT EDITIONS	14
APPENDIX 5: REMOTE INSTALLATION OF EA.....	15
APPENDIX 6: SEPARATE REPOSITORIES UNDER A COMMON MASTER REPOSITORY	16

Introduction

Enterprise Architect (EA) offers a diverse set of functionality designed specifically for deployment in large corporate environments. Given the diversity of structure in corporate environments, there is a need to allow for a number of different schemas for deploying EA. This white paper discusses the different deployment schemes as well as the features in EA that can be used to set up these different deployment scenarios for repositories.

Typical Deployment Schemas

The following schemas are common methods of deploying Enterprise Architect in a large corporate environment;

- Single site
 - One building with a large user base on a LAN
 - Multiple buildings with a high speed WAN
- Multiple sites
 - Multiple Sites with a low speed WAN
 - Multiple sites no specific WAN
- Single Site + offsite users
 - One large site i.e. Head Office with multiple external contractors working on customer sites.
- Single Site – Multiple Projects
 - Multiple repositories
 - One Repository – many Projects

Deployment Features

There are some common features in EA that are specifically designed for this type of deployment. When these features are used in combination they create a flexible environment that allows for a specific layout or one that is open ended and allows for expansion.

These features are dependant on the different version of EA installed – see the [Enterprise Architect Editions](#) for more information on what features are available on the different versions.

The following are some of the key features of EA that can be used in these processes;

- EA offers a choice of two key types of repositories:
 - EAP file repositories
 - Allows for replication hierarchies
 - DBMS Server repositories
 - Robust
 - Secure

- High volume usage
- Data Transfer between repositories
 - EAP files ⇔ DBMS Repositories
- Package control
 - Batch Import/Export of packages
- Version Control
 - Version control for use with team development
- Security
 - Secure locking of access to packages

Choice of Repository

A key feature of EA is the capability to work with a local repository or with a back end SQL DBMS repository. Both of these have there different pro's and con's for using them.

A simple comparison for using these is listed as follows;

Function	EAP	DBMS
Replication	Yes	NO
Number users	1..10	10..*
Non-Corruptible	No	Yes

EAP file Repository

A key behind the differences of these is in understanding the .eap files. These are based on the Microsoft Jet 3.5 database engine (MS Access '97 format .mdb¹) files. The inherent features of this file type include the ability to use replication. There are limitations to this type of repository when considering large numbers of users. These are;

- a) Access speeds may become slowed with a large user base
- b) The rare possibility of corruption to the repository (recoverable) if a network/power failure occurs while writing data to the file.

¹ EA supports the Jet 4.0 database engine as well, but this needs to be selected internally under the options section .

DBMS Repository

A DBMS server backend repository overcomes the two shortfalls outlined above. The DBMS servers response time to processes in a large user base are far better than the MS Access based file structure due to the inherent structure of the DBMS server backend. Further, any networking problems should be covered by the ability of the DBMS server to fully backtrack any transactions that are prone to failure caused by any external breakdown.

The types of DBMS repositories supported by EA are MS SQL Server, MySQL, Oracle9i, PostgreSQL, MSDE and Adaptive Server Anywhere. See [Corporate Resources](#)² for more details.

Deployment Schemas

The following are some details for configuring the four typical deployment Schemas outlined above;

Single Site Scenario

The corporate edition allows for two common methods for setting up a central repository. These are as follows;

- 1.1. Central .Eap file repository on a File server
- 1.2. DBMS server backend repository on a Data server.

The first is a simple scenario which can be used for groups in the range 1 to 10 users. It gives a straight forward system whereby a repository is stored on a file server that is accessible across the LAN by users of EA. All the key features of EA corporate edition are available using this schema. There is further scope to allow for replication to take place for any external repositories that are taken offsite and returned. As detailed above, the limitations of this schema consist of;

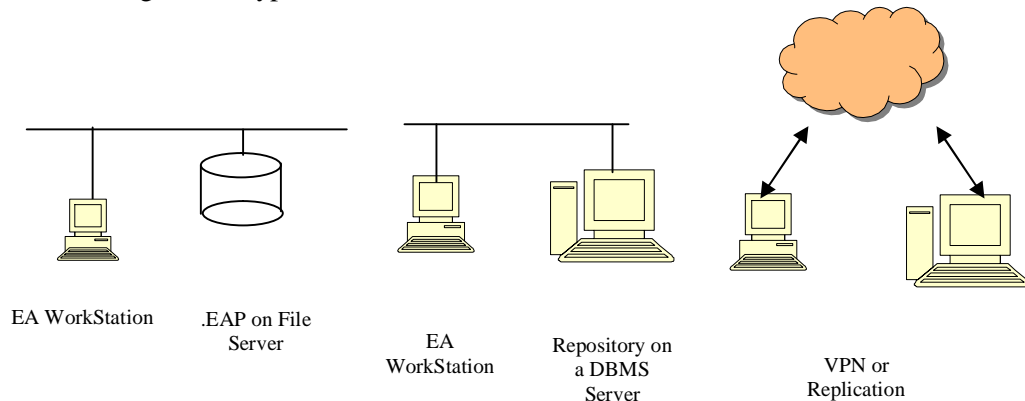
- a) Slowness due to excess user interaction
- b) Vulnerability to the repository caused by network or power failure

Due to the inherent features as outlined above, the DBMS repository offers a far more robust user environment in situations where there are a large number of users.

The DBMS repositories do not offer the replication services provided with the .Eap structure. This is covered by the Package control and Version control facilities available in the Corporate Edition. See: [Replication versus Package/Version Control](#).

² http://www.sparxsystems.com.au/ea_corp_ed.htm

The following are the types of schemas;



Multiple Site Scenarios

- Multiple Sites with a low speed WAN
- Multiple sites no specific WAN

This type of scenario can consist of a central repository with multiple sites accessing and updating the central repository. It is assumed that the network connectivity for this type of scenario does not allow for immediate updates to the central repository. Hence there is a need for the off-site repositories to have the ability to update periodically.

There are several methods for dealing with this;

- a) One central .eap file with replication to all offsite .eap files
- b) One central .eap file using a Terminal Server emulation on all offsite machines.
- c) A central DBMS repository with DBMS repositories on the sites that use either;
 - a. Batch Import/Export of XML files
 - b. Version Control
- d) Use central DBMS repository - offsite users work with eap master/children and use either;
 - a. Batch Import export for updates to the central repository
 - b. External third party replication from DBMS to MDB (.eap files)

A detail description of these schemas is as follows;

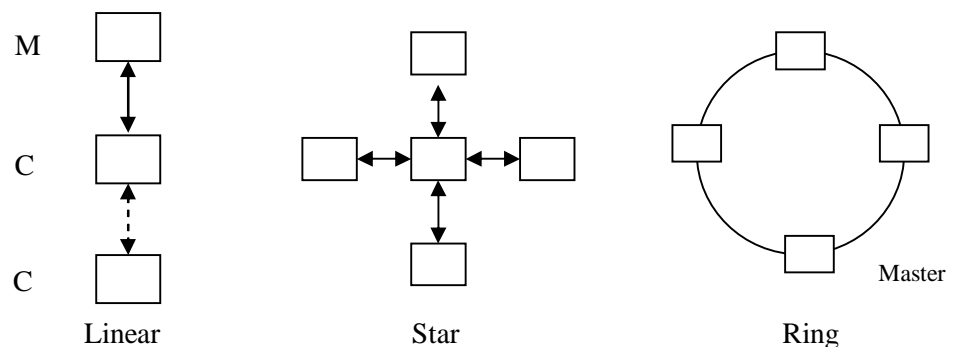
1.3. Central EAP file

In a scenario where there are multiple sites doing work on a project with defined sections such as; organizational units and one central unit that is largely involved in compiling the organizational units and producing reports etc., then the scenario

of using a master/slave is an option. It would require that the number of users in each of the organizational units is the range of 1 to 10.

The central unit then has to set up a master repository with a set of slave repositories. This can be in a number of different patterns (See: MS white papers - [Jet 3.5 White Paper](#) , [Replication White Paper](#),³).

The replication can be in the form of *linear, ring or star Patterns*. Some of these types of patterns are shown in the following diagram;



The typical pattern is one master with one or multiple slave repositories. See - [MSDN replication](#)⁴ for more details on these patterns of replication.

1.4. Use Terminal Server Emulation

In cases where the connection speed between sites is fast (20 – 60 ms) but the volume of throughput data is restricted (i.e., a 64k bandwidth), then a terminal server type application can be used to emulate EA on the remote work stations.

1.5. A Central SQL repository

With DBMS repositories the replication functionality is not available. The key function which replaces this is the ability to set up Package Control and Version Control.

The Package control provides a mechanism for 'externalizing' parts of an EA model. By using controlled packages it is possible to support a widely distributed development by batch Import/Export of packages in the form of XMI files. This

³<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q164/5/53.asp&NoWebContent=1>
<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q181/3/71.ASP&NoWebContent=1>

⁴http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnaraccess/html/msdn_replicat.asp

allows the central EA repository to perform a batch upload or distribution to external repositories (DBMS or EAP). This can be further extended to include version control as the means of accessing the central repository. See [Package Control](#) for details on setting this up.

2. Single Site + offsite users

It is not uncommon to have one large site acting as Head Office with multiple external contractors working on customer sites. In this scenario the DBMS server is best used for the central repository. It is recommended that the offsite users use the .eap repositories. To interchange the data on these repositories it best to use the Version Control or Package Control.

If there is no need to set up version control then package control can be used to enable a batch process of import/export to be performed. If version control is used, it will allow users to check-out and check-in using the inherent locking associated with the version control.

3. Single Site – Multiple Projects

In a scenario where there are multiple projects under development it is possible to set up multiple repositories or if there is common information to be accessed between projects it is possible to set one main repository with multiple Root Nodes or multiple Views. Each root node or View can contain a separate project.

In the second scenario Version Control can be useful to ensure the main repository is kept intact without excessive temporary or unwanted user information. This way the user would be using Version Control to export parts of the repository to their own local repository (this can still be a multi-user local repository). This would be then loaded back onto the main repository on completion of work.

The full local repositories can be completely updated from time to time to reflect any functional changes on the central repository see [Using Separate Repositories under a Common Master Repository](#).

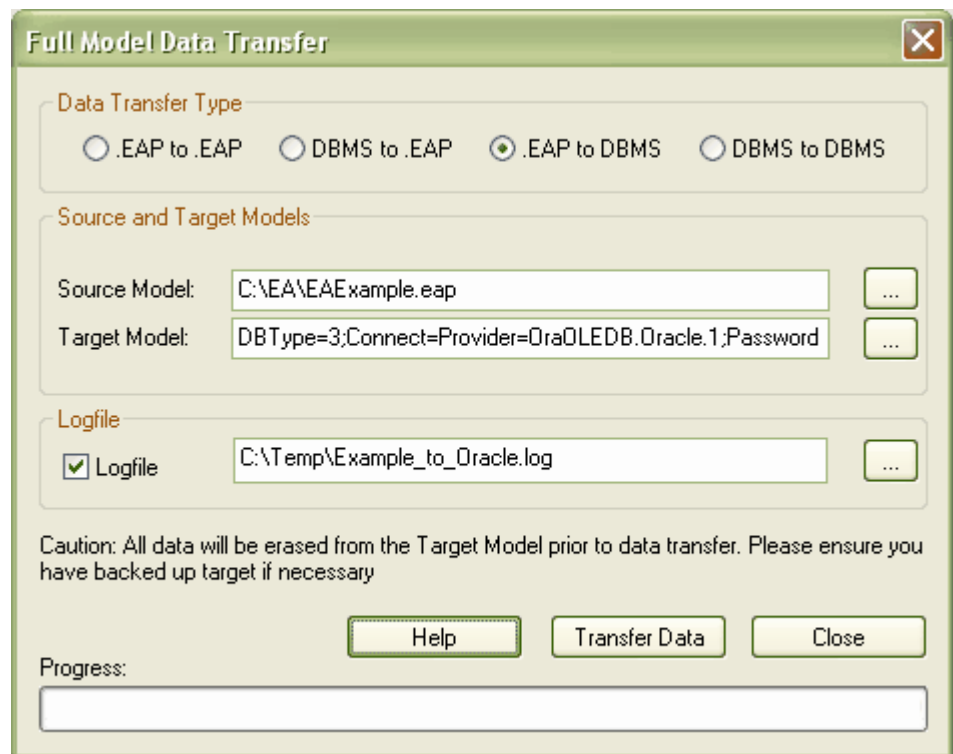
[Security](#) is also a useful feature in this type of scenario – to ensure access to the different projects is kept within the bounds of the different user groups.

Interchanging Data between .eap Files and DBMS Repositories

A key process to use when setting up an offsite repository is the interchanging of data between different repositories. This is supported in EA with the Data Transfer function. This provides the mechanism for interchanging data as follows;

- EAP ⇔ EAP
- DBMS ⇔ EAP
- DBMS ⇔ DBMS

This can be accessed from the main menu of EA under **Tools | Data management | Data Transfer**. This will open the *Full Model Data Transfer* dialog:



Common scenarios for using this process are in the initial set up of a DBMS repository. This process can also be used for transferring data to different repository types when data is interchanged between repositories.

Appendix 1: Replication vs. XMI Import/Export

As stated above the two key features available in EA for distributed development are the replication of eap repositories and the use of XMI (used in Package and Version Control) to interchange data between repositories.

Replication is a simple process for ensuring valid interchange between repositories. When using this, geographically separated analysts can update and modify parts of the model in replicas, then merge these back together at a central location. As stated earlier, replication as an application feature is restricted to the .eap repositories. For more information on the process of replication setup see under help in EA: Replication.

When using a DBMS (or a large .eap file) as a repository, the XMI based Import/Export facility can be used to model discrete packages, export to XML and share these among the development team. This approach has several benefits over replication:

1. You can assemble a model from only the parts you need to get your job done
2. You can assemble a full model if required.
3. You can assemble a model from different package versions for different purposes (eg. customer visible, internal release only etc.).
4. You can roll-back parts of a model on an as required basis.
5. There is less chance of 'collisions' between developers if each works on a discrete package
6. The process is controllable using a [Version Control](#) system.

For more information on how to set up bulk import/export - see [Package Control](#) below.

Appendix 2: Package Control and Version Control

EA provides two different methods for version control. These are;

1. Package control
2. Version control

Although both versions are similar in that they can be used for coordinating the sharing of packages between users. Version control gives the added feature of saving a history of changes to EA packages, including the ability to retrieve previous versions.

Some more details on package and version control are as follows;

Package Control

To set up the simple batch import/export of a set of EA packages from a branch office to a central location such as the head office repository, it is necessary to set up a set of packages as 'Controlled Packages'.

The process for doing this is defined in the EA helpfile under: Configure Controlled Packages. Once this has been completed you then have access to use the batch import/Export as follows;

1. From the *Project* | *Import/Export* submenu, select *Batch XMI Export*.



2. In the *Batch Export* dialog, check the packages to include in this export run.
3. Press *Save Settings* if you wish to save this configuration as the default.
4. Press *Run Export*.

Version Control

EA's Version Control allows for the repository to be maintained by a third-party source code control applications that controls access and record revisions.

It can be set up using source code control software that is SCC compliant such as Microsoft Source Safe, or CVS compliant. There numerous packages available that are SCC or CVS compliant or have conversion tools to be SCC or CVS compliant.

There are numerous advantages that can be gained by editing a shared model, and checking it into a source code control. These are dependant on the environment and the type of process needed such as support for different releases and versioning.

There are four basic ways in which the version control facility can be employed:

Use	Description
Shared model	Users sharing a central EAP file or SQL database. This configuration allows users to see other users' packages without explicitly having to retrieve them. Version control regulates access to packages, and maintains package revision history.
Duplicate models	An EAP file is created by a single user who configures it for version control. The file is then distributed to other users who connect using their own user name. Other users' packages are retrieved using the Get Package command
Shared packages	Individual users create their own EAP files but share one or more packages by connecting to the same SCC project and using the Get Package command
Standard packages	A company may have a standard set of packages which are broadly shared (on a read-only basis). Individual users use the Get Package command to connect to the main package.

For more information on the set up of the version control systems in EA – refer to the helpfile under:

- Model management
- Version Control.

Appendix 3: Security

EA provides security. This limits access to update functions in the model. Elements may be locked on a per-user or per-group basis and a password is required to login.

EA provides two possible security policies which are either:

a) Standard Security Model

All elements and diagrams are considered unlocked, and as the need arises, a user may lock any element or set of elements at the user or group level.

b) Rigorous Security model.

This assumes everything is locked until explicitly checked out with a user lock. In this mode, an EA model is read-only until a user applies an editing lock to one or more elements.

Security in EA is not set up for stopping unauthorized access; rather it is intended to provide a means of improving collaborative design and development by preventing concurrent editing and limiting the possibility of inadvertent model changes by users not designated as model authors.

Security allows elements to be locked for change but still makes these accessible to users to place them as linked items in diagrams where the user has access. These elements will shown in the diagrams but not able to be changed. An example of this is where a definition of a server is locked by the architect but is able to be displayed in a diagram created by the deployment manager.

For more information on security features in EA see Help in EA under:

- Model management
- Team Development
- User Security

Appendix 4: Enterprise Architect Editions

Enterprise Architect is available in three editions - **Corporate**, **Professional** and **Desktop**. Functionality for each edition is as follows:

Features \ Edition	Corporate	Professional	Desktop
.EAP Files	Y	Y	Y
Shared Models	Y	Y	N
Source Code Engineering	Y	Y	N
Database Engineering	Y	Y	N
Microsoft Access Repository	Y	Y	Y
MS SQL Server, MySQL, Oracle9i, PostgreSQL, MSDE, SQL Anywhere	Y	N	N
Version Control	Y	Y	Y
Replication	Y	Y	N
MDG Technologies	Y	Y	N
Security	Y	N	N

Corporate Edition

The Corporate Edition is aimed at larger development teams. It supports everything in the Desktop and Professional versions, as well as the ability to connect to SQL Server, MySQL, Oracle9i, PostgreSQL, MSDE, Adaptive Server Anywhere and MS Access backends as the shared repository. It also supports user security, user logins, user groups, replication and user level locking of elements. Support for MDG Technologies is included with the Corporate version of EA. Additional resource are available to users of the Corporate Edition.

Professional Edition

The Professional Edition is aimed at work groups and developers. It supports shared projects through replication and shared network file. The Professional version also has an ActiveX interface for interrogating EA projects and extracting information in XMI format. The Professional version also fully supports synchronization reverse engineering of code as well as import and export of database schema. Support for MDG Technologies is included with the Professional version of EA.

Desktop Edition

The Desktop Edition is targeted at a single developer producing UML analysis and design models. It includes all Professional features except code engineering (import/export of source code and DDL), the Active-X interface and the ability to share a model amongst multiple users.

Appendix 5: Remote Installation of EA

When deploying EA across a network of workstations there are a number of applications and methods for deploying an application in this way. This can be done using Windows Server tools or the likes of SMS. The following just gives a brief explanation of the basic method for creating an MSI for use in this type of remote installation.

Regarding setting up an EA MSI installer – the setup.exe does include a windows installer MSI installation but it is embedded inside the setup.exe. On running this exe file it extracts the .msi and verifies that the host PC has the correct version of Inst MSI. Once the MSI is extracted it can be used to carry out the remote installation of EA.

The method to separate out the MSI for your remote installation is as follows;

1. Double click on EA's setup.exe
2. Go through the steps just passed acknowledging the license.
3. Browse to %Temp% or C:\Documents and Settings\Administrator\Local Settings\Temp. *
4. Find the MSI file - [RandomFilename.msi](#) (best to just sort by date and go to the latest entries). Copy this MSI file to a different location.
5. You can then use this file to do the installation through Windows Server or SMS etc.

* Notes:

1. {administrator} may be the current username - if you have a separate administrator UserName use the directory named as the UserName you have entered.
2. You can also find that the .msi extracted by the setup is in usually in %temp% with a random filename. If you run the EA setup up to the point where the first screen of the wizard appears you can then take a copy of the .msi in %temp%.

Appendix 6: Separate Repositories under a Common Master Repository

In some scenarios it may be best to keep a common repository with different Project Roots for each application as well as one Project Root for the common functionality. In other cases it may be best to keep these separate but update the individual application repositories on a periodic basis.

When using multiple repositories with a common system architecture, EA has a feature that can be used to propagate the key aspects of the repository without the core data. This can be carried out by doing an export of the most up to date structure in a master repository and import this into the repositories for the related applications being developed.

Some of the core parts of the repository that may need to be propagated are:

- Copying glossaries from one model to another
- Adding additional stereotype profiles by merging new stereotypes into the model
- Copying resources, clients, etc. from one model to another

When reference and project data is exported, Enterprise Architect writes it out to a custom XML file. This includes table information, filter information, rows and columns.

To export data, follow these steps:

- 1) From the *Tools* menu, select *Export Reference Data*.
- 2) In the Data Exporter dialog, select the table(s) you wish to export - you can select one or more tables for a single file.
- 3) Press *Export*.
- 4) Enter a valid file name with a .XML extension when prompted to do so.
- 5) This will export the data to the file.

This can then be imported into the related repository using *Tools* menu, select *Import Reference Data*.