Enterprise Architect

User Guide Series

Deployment

A detailed guide covering the deployment options for Enterprise Architect.

Author: Sparx Systems

Date: 8/04/2021

Version: 2.0

Table of Contents

Introduction

Sparx Systems' Enterprise Architect is a comprehensive, scalable modeling platform designed for use within large corporate environments, providing a range of deployment options to accommodate the unique needs of modern organizations. This broad set of features can be configured in multiple different combinations, giving a wide range of options for you to create your own specific layout that is open for further expansion.

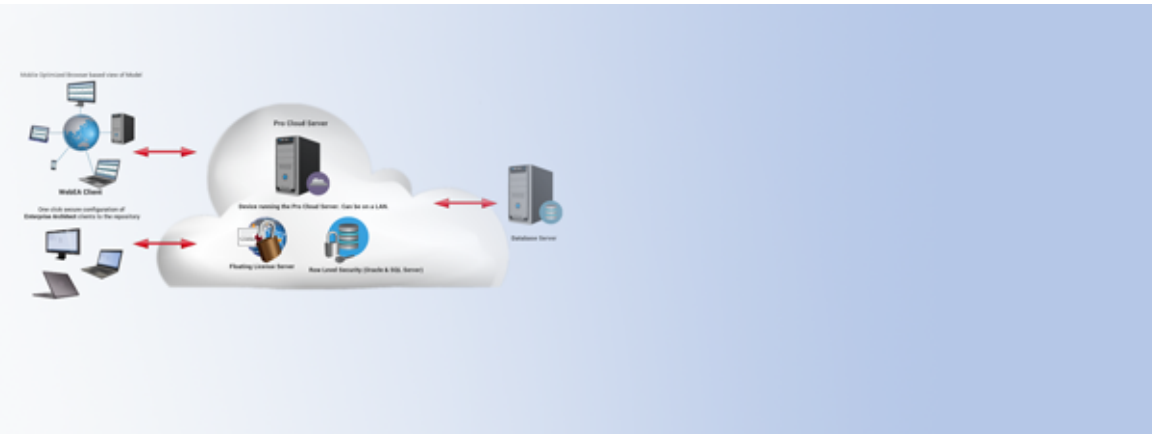
This white paper provides a general overview of deployment scenarios, ranging from a simple office to large multi-site organizations. It covers the variety of options available and how they can be used in combination for optimal performance and ease of use.

The content has been grouped into these sections:

- **Deployment Scenarios**
  Given the diversity of deployment options available for Enterprise Architect, this section runs over a range of deployment scenarios used in typical organizational structures.
- **Deployment Options**
  When configuring your own environment it is crucial to be aware of all the options available; this section discusses the core features for configuring your specific deployment of Enterprise Architect, including choosing the optimal Repository type, Network deployment options, client/stakeholder access to review and provide feedback, as well as features useful for multiple large on-going projects.
- **Change Control**
  Enterprise Architect supports a range of different options for managing Change Control, which can be critical in a team based environment; this section covers Auditing, Time Aware modeling, Baselines, Re-usable Asset Service (RAS) and Version Control.
- **Team Collaboration**
  Collaboration between team members is a critical aspect of working with a modeling system and striving to achieve a productive outcome; this section discusses the core features for team collaboration and the roles that they can play in your deployment scenario.
- **Set up**
  The set up section provides a general overview of the processes and resources required to configure the different deployment scenarios; it covers the standard workstation installation options, remote installation and a simple Pro Cloud Server installation.
- **Optimization**
  When sharing large models over a network or a Cloud connection you must consider choosing the most

appropriate configuration; this section looks at a range of repository and networking options that can be used in optimizing your deployment scenario.

Deployment Scenarios



Enterprise Architect provides numerous deployment options to accommodate the variety of organizational structures in which is it used, ranging from large multi-site organizations through to single site developer groups.

These sections provide a number of core deployment scenarios grouped as:

Multi-site

Using Pro Cloud Server there are numerous options for multiple-site connectivity. This is an overview of common configurations:

- Multiple sites with a Cloud connection
- Head Office with external contractors working on customer sites
- Project input interaction
- Small intercontinental teams
- Outsourced teams
- Amazon/Azure

Single site

Single site options include:

- On-site Cloud Connectivity
- One building with a large user base on a LAN
- Small team using a file based repository.

Client Exchange

There are a number of options for distributing your model to your client base. These options include:

- Prolaborate
- WebEA
- Joomla!
- Web reports
- EA Lite

Multi Site

When connecting multiple sites to a common repository a fundamental issue is in dealing with a communication bottleneck between the sites. When connecting directly to a DBMS across a WAN, this can cause serious delays and unmanageable response times. The Sparx System's Pro Cloud Server (PCS), optimizes this by using a HTTP/S connection to exchange compressed data between the workstation and the DBMS.

With multi-site connectivity there are numerous scenarios. Some of the common patterns that use the PCS include:

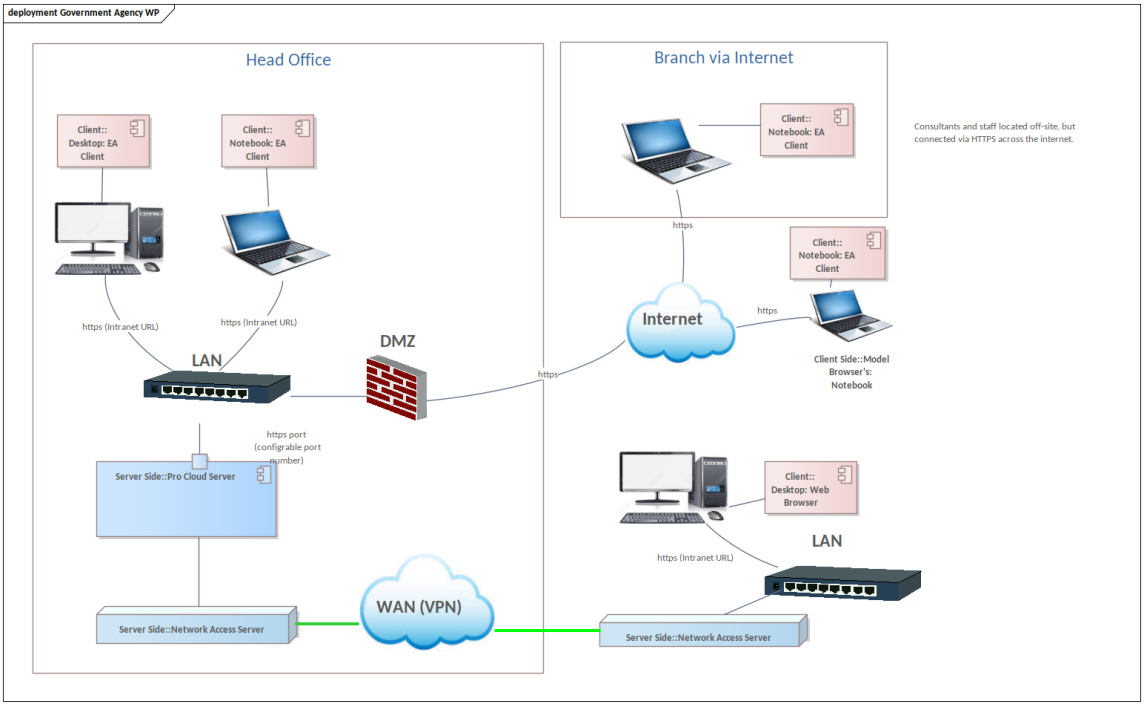
- Multiple sites with a PCS connection
- Head Office with external contractors working on customer sites
- Small intercontinental teams (standards Development)
- Outsourced teams
- Amazon/Azure connections

The following sections outline possible configurations covering these scenarios.

Multiple sites via Pro Cloud Server

Using the Sparx Systems Pro Cloud Server you can start with an on-site HTTP(S) connection to your repository. This can easily be extended to include off-site connections to branch offices or outsource vendors.

The external connections can be via simple HTTPS connections (via firewall), or using more dedicated VPN style connections.



This scenario provide a HTTPS connection from the workstations, in the head office and branch offices, to a repository situated in the head office.

Notes:

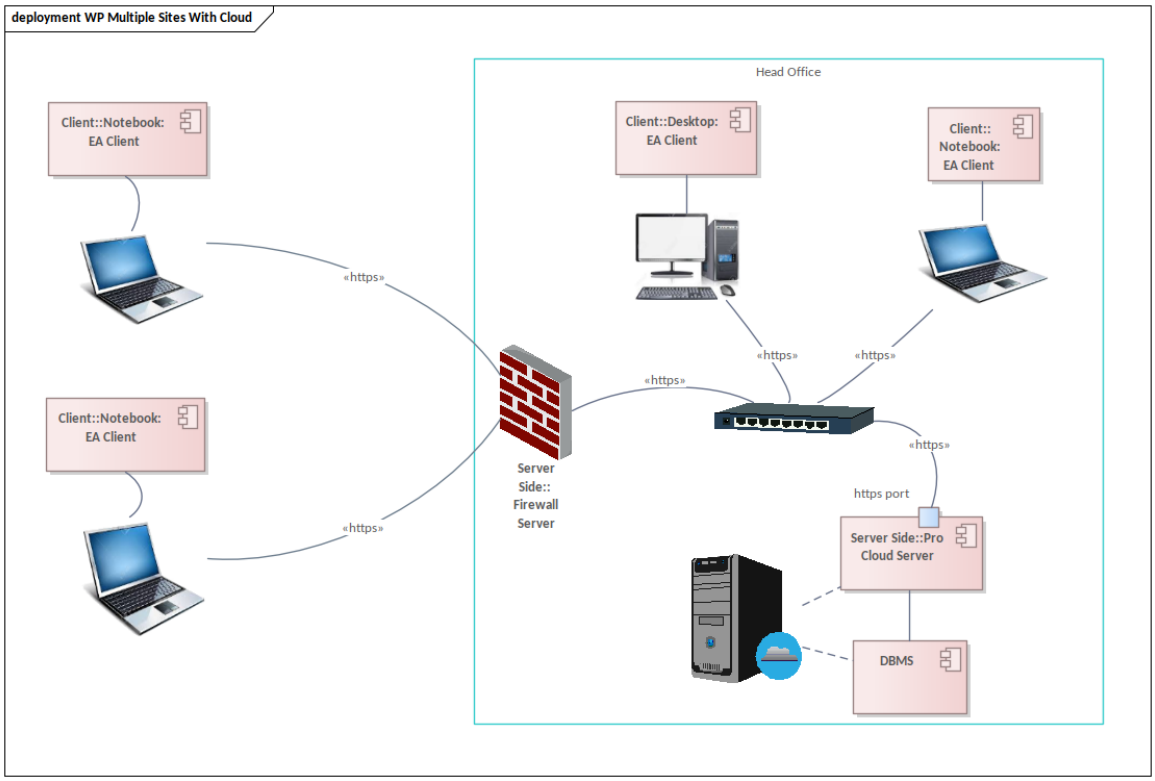
- This assumes a branch office's WAN connection to the Cloud server has a latency of lower than 100 ms (preferably <50ms); if the network latency is higher than this there can be issues with speed
- This can use either a Firebird file based repository or a DBMS repository; a DBMS repository would be required for larger teams (any of 10+ DBMS options)
- Where Visibility Level security is required (Row Level security), this will narrow the repository options to either MS SQL or Oracle

Advantages:

- Off-site direct access to the repository ensures everyone has immediate access to the latest information

Head Office with external contractors

The Sparx Systems Pro Cloud Server provides a platform for intranet style HTTP/S connection to your internally hosted DBMS repositories, which can also be extended to external HTTPS access to allow for off-site usage, by simply allowing Port access via the firewall.



Notes:

- Off-site access can be a direct connection to LAN via a firewall or via a VPN connection
- This can use either a Firebird file-based repository or a DBMS repository; a DBMS repository would be required for larger teams (any of 10+ DBMS options)

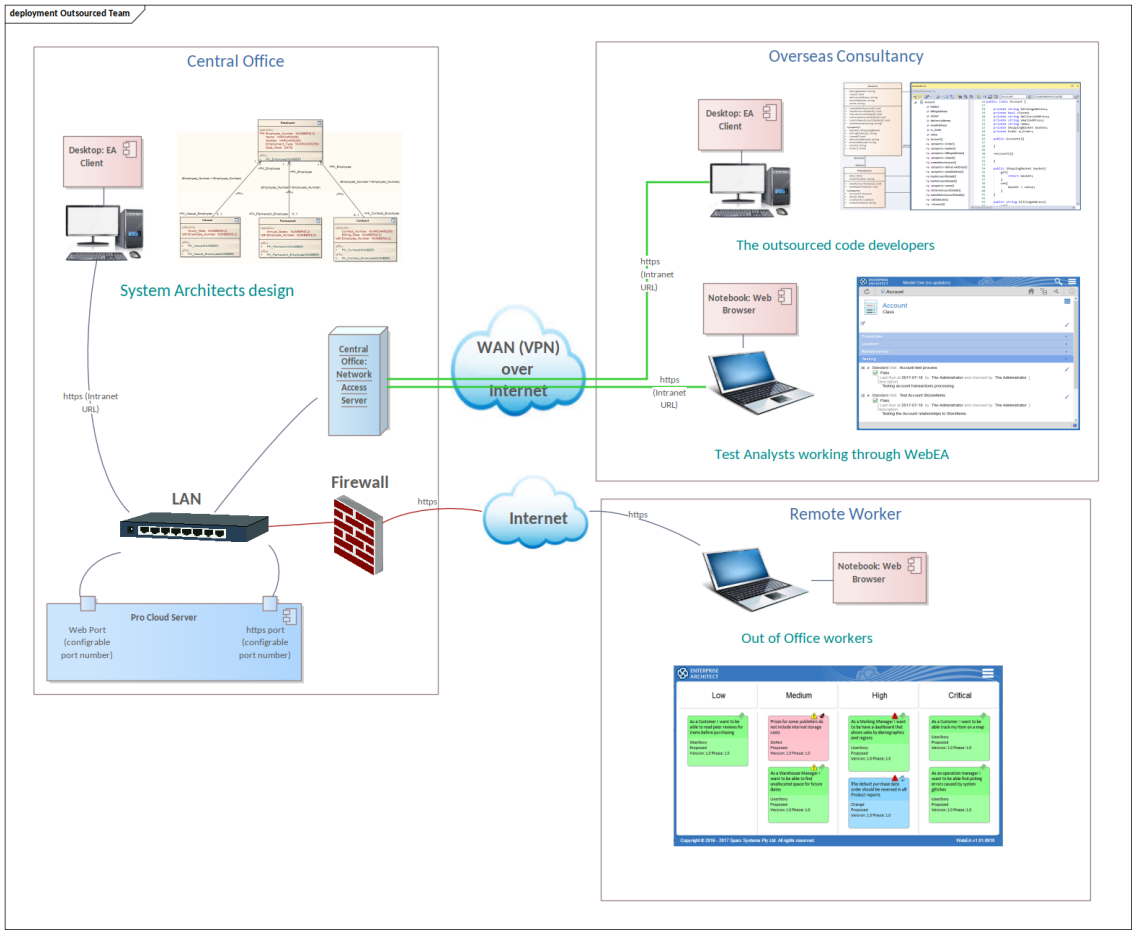
Advantages:

Off-site direct access to the repository ensures everyone has immediate access to the latest information.

Outsourced teams

A common scenario for using the Pro Cloud Server is where IT developers work with an outsourced team for code production. The Central Office relies upon developers who either work remotely or provide services from one or more consultancies located in other countries. Some individuals and organizations exchange data through a secure connection across the Internet, others connect via a Virtual Private Network on a WAN.

The Pro Cloud Server unlocks the models that were traditionally only accessed by staff in the Central Office via an Enterprise Architect Client, to anyone with a Web Browser, an Internet connection and the permission to access the content.



Central Office

The System Architects, in the Central office, design the detailed high level model architecture and data structures ready to be used by the outsource developers as a basis for starting the code development.

Outsourced Code developers

The outsourced code developers are working with the architectural model and requirements via a Cloud connection to the model repository. Using security, the Central office administrator has this group restricted to only viewing the design, but permits them to work with updating the code and performing visual execution of the code being developed.

Test Analysts

Test Analysts working with test design and test automation tools can use WebEA to view the requirements, business rules, and test cases in these models through the Web Browser.

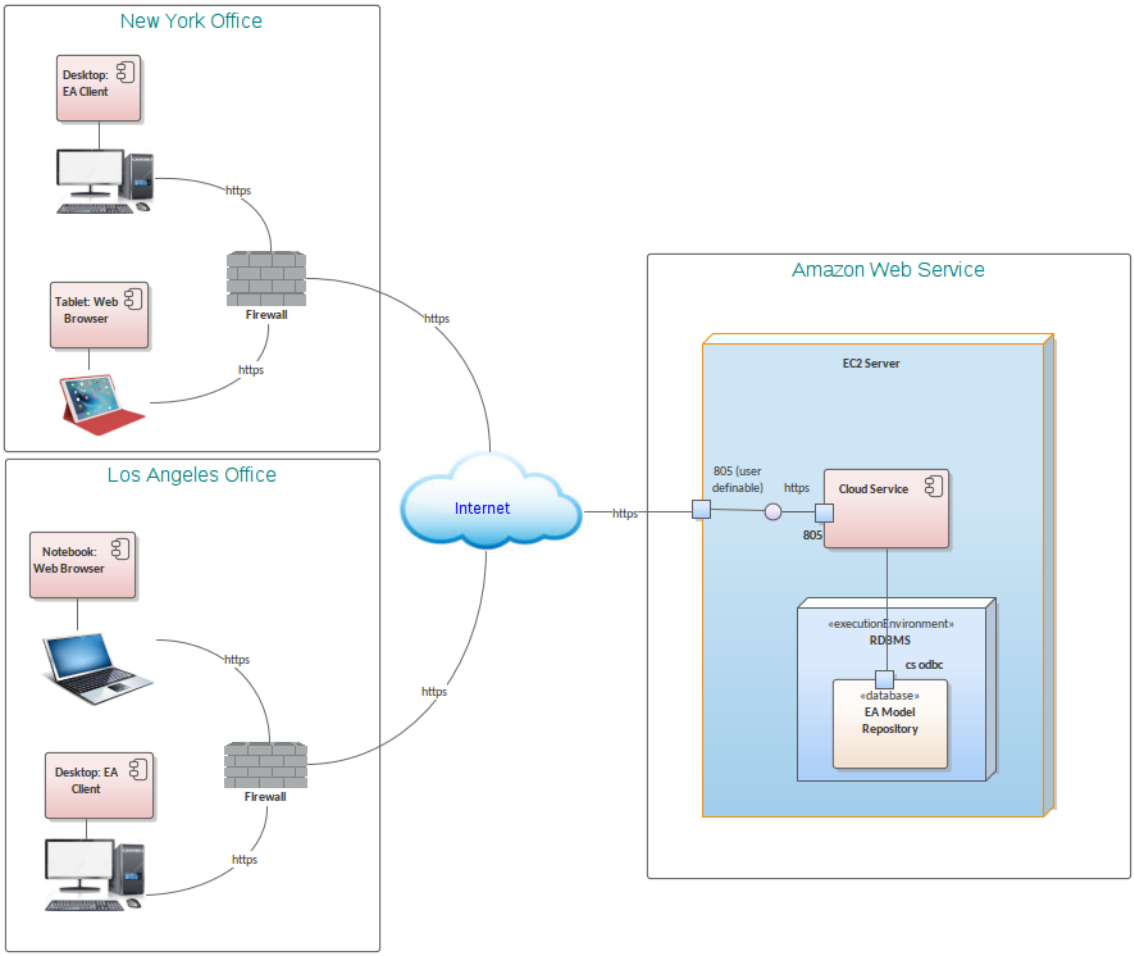
Out of Office workers

Out-of-office workers can use WebEA to view and follow discussions, view Kanban diagrams and, if configured to do so, update Requirements notes and Issues.

AWS

The Sparx Systems Pro Cloud Server can be easily deployed on a 3rd party cloud service such as the Amazon Web Service (AWS). If you place your repository on AWS, then performance will be slower than on your LAN. The Pro Cloud Server provides compression and parallel flow to give comparable performance to a LAN connection, so long as there is reasonable latency between the workstation and AWS sever.

A common set-up scenario on the AWS is to use an EC2 server to host both the Pro Cloud Server and the DBMS.



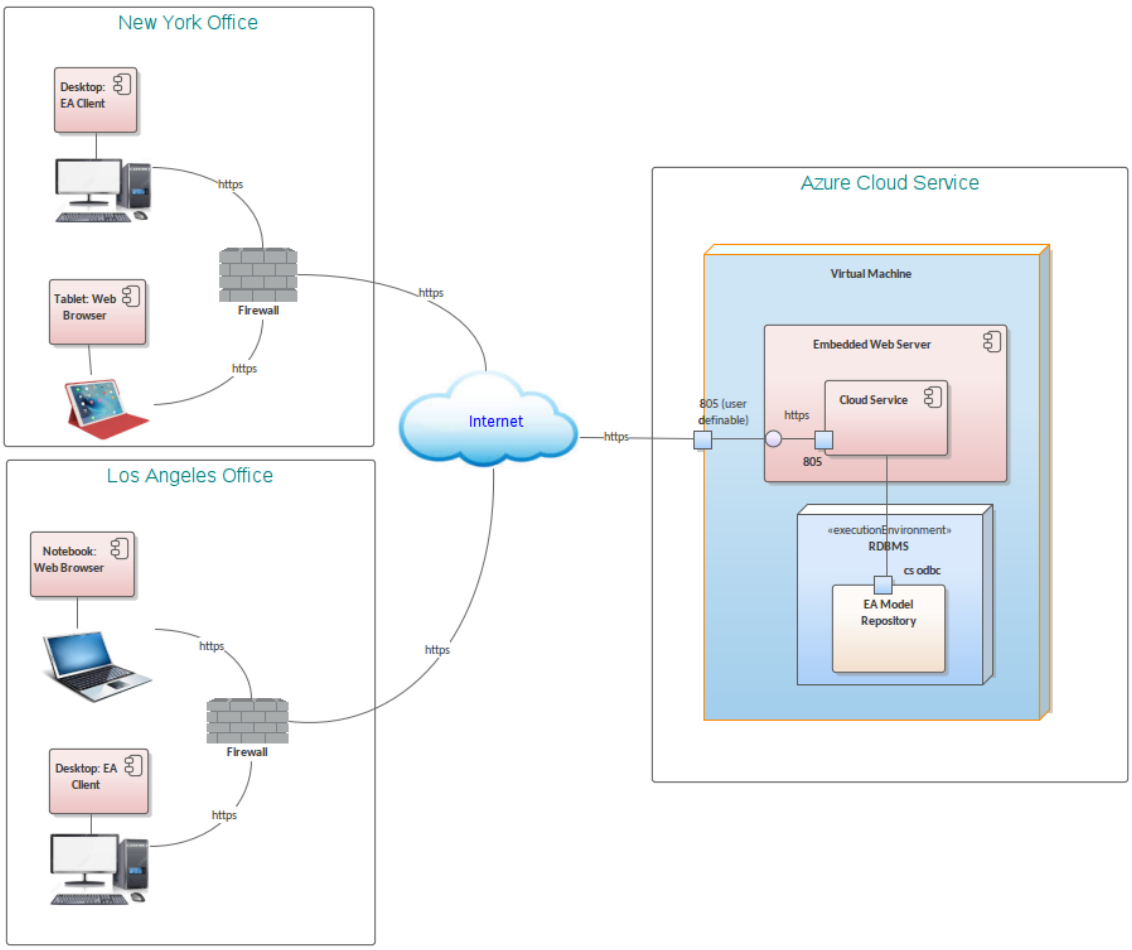
Notes

- The Pro Cloud Server ports are configurable and can be set to different port numbers to those shown in the illustration.
- Ensure that the network latency between the workstations and the AWS server is within a reasonable range <100 ms - preferably <50 ms. Simple ping testing of the workstation connection requires that ICMP is enabled on the EC2.
- The EC2 server has internal firewalling that must be configured to allow external access to the Cloud service port(s).
- AWS also provides Appstream 2.0 hosting that you can use to deploy and stream the Sparx Systems Enterprise Architect desktop application for your company for a fully managed, secure application streaming service that runs in the AWS Cloud and is accessible via a web browser.

Azure

The Sparx Systems Pro Cloud Server can be easily deployed on a 3rd party cloud service such as the Microsoft's Azure Cloud service. If you place your repository an Azure, then performance will be slower than on your LAN. However the Pro Cloud Server provides compression and parallel flow to give comparable performance to a LAN connection, so long as there is reasonable latency between the workstation and the Azure server.

A common set-up scenario on the Azure web service is to use an Azure 'Virtual Machine' server to host both the Pro Cloud Server and the DBMS.



Notes:

- The Pro Cloud Server Ports are configurable and can be set to different Port numbers to those shown in the illustration
- Ensure that the network latency between the workstations and the AWS server is within a reasonable range <100 ms - preferably <50 ms. Simple ping testing of the workstation connection requires that ICMP is enabled on the EC2.
- Simple ping testing of the workstation connection requires that ICMP is enabled on the Virtual Machine
- The Virtual Machine server has internal firewalls that must be set to allow access to the Pro Cloud Server Port(s)

Single Site

A common scenario with a large organizations is to have one building housing the core staff, with a large user base on workstations that connect via a local area network (LAN). For an organization that primarily uses Enterprise Architect across a standard LAN there are three core options available:

- HTTP(S) access via an on-site Pro Cloud Server
- Access to a DBMS using OLE DB or ODBC
- Small teams using a file based repository

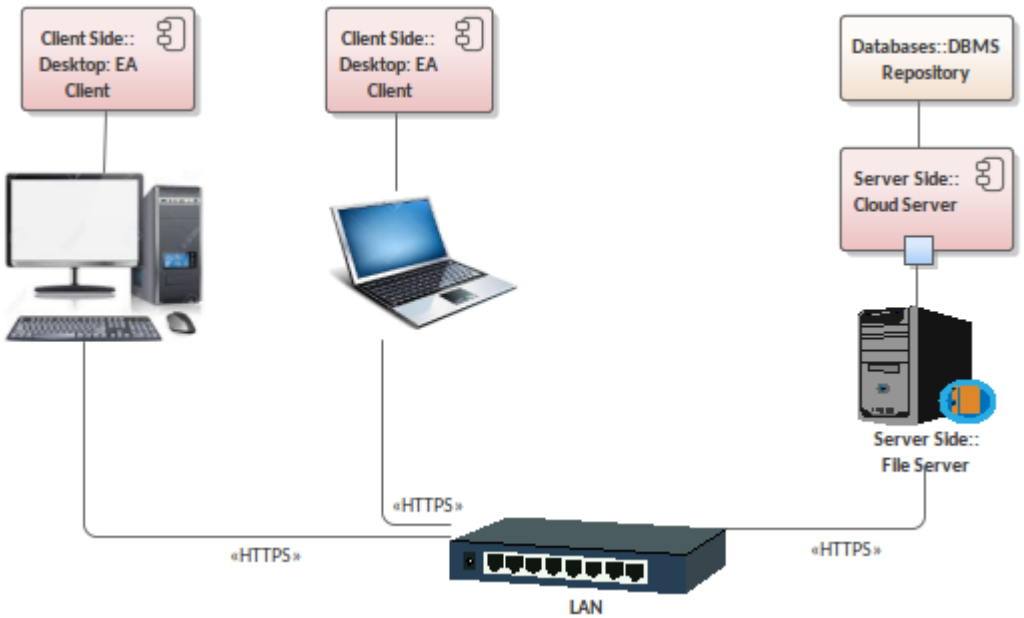
The next section provides clarity on the configuration and the benefits and shortfalls of each option.

On-site Cloud access

The Sparx Systems Pro Cloud Server provides a platform for intranet style HTTP/S connection to your internally hosted DBMS repositories.

Advantages:

- Supports DBMS repositories
- Leaner network data throughput
- Simple connection - ODBC/OLE DB drivers are not required on each workstation
- HTTP ports can be secured to remain within the local Network
- Allows for optional off-site connectivity

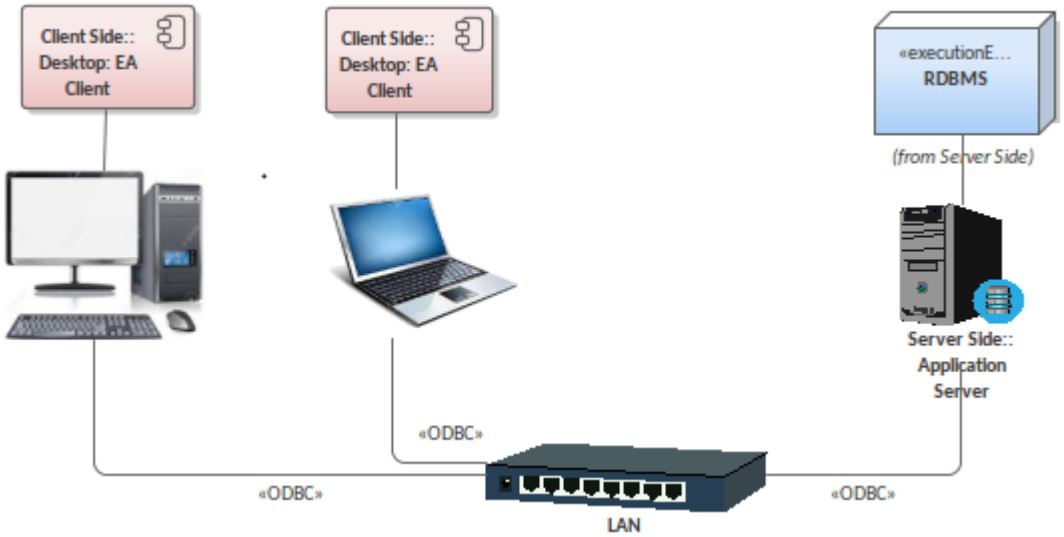


When using Pro Cloud Server you have a choice of using a Firebird or a DBMS repository. A DBMS repository is preferred for a large user-base. For a smaller user base you can use a file based Firebird repository, which can be a good starter for a small but expanding organization.

One building with a large user base on a LAN

When using multi-user accessible repositories, with the core users working in one building, you can opt to use direct workstation-based ODBC connections to the repositories on a DBMS server. Although the Pro Cloud Server HTTP access is a simpler option, for a medium sized single-site organization there might be preference not to use a Cloud server. The reasoning behind using the older direct ODBC / OLE DB connection to a DBMS repository is:

- Where the user base is over five users it is recommended that a DBMS repository is used not an .eap file
- Multi-user access to a Firebird file based repository, that supports a larger user base and larger file, is only available via the Pro Cloud connection



Advantages:

- No limitation on repository size

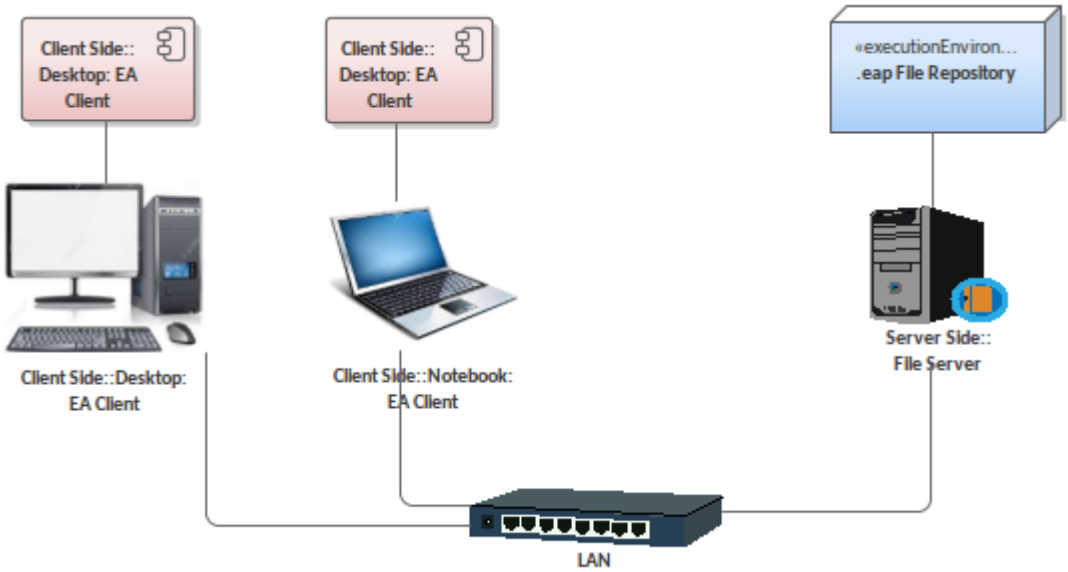
Limitations:

- Restricted to a low Network latency LAN connection
- Requires ODBC /OLE DB drivers be configured on each workstation.

Small team using a file based repository

File based repositories can give you quick access to create a simple mock-up model, or they can be used for small teams of fewer than 5 users for creating models in a team-based scenario.

For interactive usage a .eap file is commonly stored on a file server accessible to the team.



Advantages:

- Very simple set up

Limitations:

- Supports only small groups of concurrent users
- There is a limit to the size of repository (less than 50 mb)
- Database is prone to corruptions

Client Exchange - Exchanging and Publishing content

Enterprise Architect provides a range of options for interacting with the client/stakeholder base, as well as different means of publishing designs for review and for implementation.

When considering the configuration of Enterprise Architect in a corporate environment, you might want to use a combination of these options:

Prolaborate

Prolaborate is designed to make model information accessible to the high-end users of the organization, who are overseeing the modeling. It can be easily tailored to provide a set of views that reduce complexity, focus the attention on viewing the direction of the project, and provide clarity on whether the project is achieving the desired outcome. This includes using dashboards, impact analysis, gated reviews and much more to present core information to a custom audience.

WebEA

Sparx System's WebEA is a web-based interface to Enterprise Architect's models, through which users can review and comment on a model in real time. Being web-based, it operates on a range of mobile devices or on a remote work station, without any need to install Enterprise Architect.

Benefits:

- Can be used on any machine that has a web browser
- Content is dynamic, providing a real time view of the model
- Provides very simple access to model details
- Provides optional update to texts such as Discussions, Element Notes and test results
- Supports searching the model
- No publishing process required

Publish Joomla Articles

Enterprise Architect provides a facility for publishing model content as HTML Articles in a local Joomla! installation.

Benefits:

- Allows the HTML model to be searched

- Web discussion can be added

Shortfalls:

- Content is static after publishing
- Requires manual publishing of content
- Discussion comments are external and not posted back in the model

HTML Reports

You can also make your model available to others in a read-only format by generating an HTML report on the model, which can be published on the web with read-only access. This is the simplest reporting in HTML format.

Benefits:

- Simplest cost effective web based publishing

Shortfalls:

- Does not support searches on the model
- Does not support adding comments
- Requires manual publishing of content
- Content is static after publishing

EA Lite

Enterprise Architect Lite is a free, read-only edition of Enterprise Architect that can be used by clients and stakeholders for reviewing the project.

Benefits:

- Provides a real time view of the model
- Allows clients to pass feedback using Discussion and Team Library
- Does not require a publishing process

Shortfalls:

- Users need to be familiar with the Enterprise Architect user interface and menu system to access model details
- This requires a fixed installation on a workstation (Windows, Linux or Mac)

See also

- Prolaborate
- WebEA
- Publish to Joomla!
- HTML Reports
- The Read-only 'Lite' Edition

Deployment Options

Enterprise Architect supports a wide variety of options for a diversity of deployment layouts. As it is a workstation-based application it can simply run without the need for a server based environment. However, for large scale deployment, there are numerous server-based options available including:

- A selection of DBMS based repositories
- A variety of connection options
- Options for sharing resources across multiple repositories.

When deploying Enterprise Architect in a corporate environment, you might want to use a combination of these options. Configuration for these environments can range from users updating their own file based repositories, through to very large teams interacting with a common model over numerous geographic sites.

Options

This table provides a summary of the topics covered in this section, describing the core options that, in combination, can be used to support your deployment scenario:

Section	Description
Choice of Repository 	Enterprise Architect supports a range of different repository-types each with different features. Before deciding which repository type to use it is worthwhile looking over the benefits and shortfalls of each.
Connectivity	Whether dealing with a large multi-site corporation or small but diverse group of users, there is the need to be able to work collaboratively on a common model

	across a network. There are a number of connectivity options: • Pro Cloud Server licensed • Pro Cloud Server unlicensed • WebEA • Networking DBMS repositories • Networking file repositories.
Security 	Enterprise Architect's user security can be used for defining model access rights and access to features. It also supports the identification of individual users when using the team collaboration tools. This section outlines: • Security Details • Authentication options.
Multiple Projects 	With development processes it is common to have multiple projects with different teams needing separate models, but using common resources and a common framework. This section outlines the various options supporting these scenarios. • Single repository • Multiple Repositories • Reusable Asset Service • Baseline transfer • XMI Merge.

Choice of Repository

For Editions of Enterprise Architect above the Professional Edition, models can be stored in either a file based repository or a DBMS repository. The choice of these repositories depends on the type of interaction, the number of users involved and the size of the repository.

As a general rule, .eap file-based repositories are good for small models with only a few users. For larger models and larger user-groups, a DBMS repository or a Cloud based Firebird repository is recommended.

The advantages and disadvantages for each option are discussed in the *Repository Types* table.

Repository Types

Type	Description
EAP and EAPX file repository 	Enterprise Architect Project (EAP) files, are based on the Microsoft Jet 3.5 database engine. These are of the MS Access '97 .mdb file format. This format does not support Unicode Characters. EAPX files are based on the Jet 4.0 database engine. They do support Unicode Characters. Benefits: • Simple file access across a shared network drive Limitations: • Concurrent access is limited to small groups of users - fewer than 5 users. • Data corruption can occur if there is a network/power failure while editing • There are limitations on the data size supported by the Jet database (less than 30-40 mb is recommended). Note: For more details on Multilingual Unicode support using .eapx files, see the notes on the *Startup* Help Topic.
Firebird file repository 	Firebird repositories, like the EAP repository, are file based and can be accessed from: • A local drive, as a file, for single user operation only • A Firebird DBMS server which supports a database being placed on a 'server' and multiple users access it using an ODBC DSN • A Pro Cloud Server for multi-user access. Benefits: • Firebird files are more robust than .eap files • When accessed via the Pro Cloud Server or ODBC there is no limit on the number of users • Unicode is supported by default. Limitations: • Data access is typically slower than that offered by .EAP/.EAPX.
DBMS repository 	Enterprise Architect supports five of the more popular DBMS products for creating database repositories. For any organization of five users or more users we recommend using a DBMS repository. DBMS repositories overcome a number of the limitations of file-based repositories.

	Benefits: • Typically, dedicated DBMS servers provide better response times for a larger user base than file based repositories • Most of the supported DBMS have no limitations on the size of repositories • Network errors are handled by the ability of the DBMS server to roll-back any transaction failure caused by external conditions • DBMS repositories can be accessed from a Pro Cloud Server. Limitations: • Creation of a repository on a DBMS requires more set up than file-based repositories (on the server and the workstation) • DBMS repositories can require some periodic maintenance, usually performed by a Database Administrator (DBA). For repository backups it is recommended that the DBMS's native backup/scheduling tools are used. Note that each DBMS has different response times in terms of server-side processing, as well as the time taken for the transmission of a reply across a connection. This can be due to different responses of the DBMS service and different data volume through-put for the same transactions across a network connection. As a direct comparison of responses for each DBMS-type cannot be supplied, we suggest you try a comparison on your own specific scenario. Our in-house testing has found that MySQL and Maria SQL provide the fastest response with the least data-volume through-put. As these are open source and free, you can use these as a benchmark for testing your preferred DBMS.

See also

- Startup
- File Based Projects
- Server Based Repositories

Connectivity

Enterprise Architect provides a number of options for connecting to a repository. Whether dealing with a large multi-site corporation, a diverse group of users, or even a single site scenario, there are a number of benefits in using the Pro Cloud Server. These include, better through-put of data and the simplified connection when accessing a repository.

This section covers scenarios where multiple users are working concurrently on a project and hence require a DBMS repository accessible to all users.

Connectivity Options

Option	Description
Pro Cloud Server - Licensed 	The Pro Cloud server provides access to a central repository via: • HTTP/S for workstation based users creating models • Web browsers for review and comment in real time **Benefits:** • Provides compression and a parallel flow of data for efficient interaction across web based networking • Does not require each workstation to have ODBC set up for access to the DBMS • Is not limited to a LAN connection - it supports slower WAN access. • Can be accessed by Clients/stakeholders via a web browser for model review and comment • Supports Visibility Levels for tight restriction on the repository data down to the row level.
Pro Cloud Server - Unlicensed 	The Pro Cloud Server unlicensed is available at no cost. The installer is accessible on the registered user site for users with a current subscription. It can be used for connection to DBMS repositories with the Corporate and above editions. It provides connectivity from the Enterprise Architect application to a central repository via HTTP or HTTPS. **Benefits:** • Provides efficient fast response with low network volume data • Supports connection to a repository across a LAN or a WAN **Shortfalls:** • Does not provide up-to-date web-browser based access to the model
LAN connection using OLE DB and ODBC 	The general ODBC connection was a common scenario used in earlier versions of Enterprise Architect. This has largely been superseded by the Pro Cloud Server HTTP/S connectivity. The ODBC connection can be useful in a small group of users working across a Local Area Network connection. **Benefits:** • Simplicity in setting up a repository **Shortfalls:** • Requires each client to have the necessary database driver installed and potentially a ODBC DSN depending on the DBMS in use

	• Uses higher volume network traffic for the same tasks • For reasonably fast access this is largely limited to use across a LAN connection.

See also

- Pro Cloud Server
- Introducing Cloud Services
- Server Based Repositories

Security

When laying out the deployment of Enterprise Architect, there are not only some core factors in risk aversion through setting security, but there are also side benefits in identifying individual users for team collaboration.

In terms of the core risk aversion, the key points for laying out security include:

- Who accesses what (model data and Enterprise Architect features)
- Who changes specific areas of the model
- Who assigns tasks to be performed
- Who is assigned tasks to be performed
- Who reviews a process

All of these factors are dependant on the security features defining who is logged in and what groups they are assigned to. Enterprise Architect supports a comprehensive user security model that allows fine grained locking of individual elements, diagrams, Packages and other model constructs.

Both group and individual permission sets are customizable by model administrators, providing a well-regulated and controlled space in which competent and trusted modelers can work alongside less competent modelers or model consumers.

This can be further tightened to set restrictions on visibility of specific areas of the model, using Visibility Levels that set restrictions that are strictly enforced on the DBMS (below the level of the application).

Security Details

Enterprise Architect Corporate and extended editions provide two forms of user definable security:

- Standard Security
- Row-Level Security (via Visibility Levels)

Standard Security

Standard Security allows for restrictions of user access to the model update functions. A password is required to log in, and elements can be locked by a user or a user group.

The standard Security in Enterprise Architect is not designed to prevent unauthorized access; rather it is a means of improving collaborative design and development by preventing concurrent editing, as well as limiting the possibility of inadvertent model changes by users not designated as model authors.

Standard Security allows elements to be locked for change; however users retain access to place them as linked items in diagrams where they have write permission. These elements will be shown in the diagrams,

but are not editable.

Visibility Levels

There is a further extension to the standard security with Visibility Levels. This provide a complete denial of access (including via SQL query) unless the user has security access rights. This utilizes the features of Oracle and SQL Server to protect sensitive content from access by anyone other than those users with security based, access privileges.

Visibility Levels is only available for the Pro Cloud server and requires that the repository is of either MS SQL Server 2016+ or Oracle 8+.

Authentication - Single Sign On (SSO)

Enterprise Architect supports trusting two forms of third party authentication for performing a security-based log in to a repository. These are:

- Windows Active Directory Authentication
- OpenID

Active Directory

Active Directory Authentication allows a model to trust the currently logged in Windows user. If the username is authenticated by Active Directory, then that user is logged into the model with the access permissions set according to the model security.

OpenID

OpenID is the current preferred standard for SSO authentication for web sites. Enterprise Architect trusts the authentication returned from the OpenID SSO system and logs the user in to the model.

See also

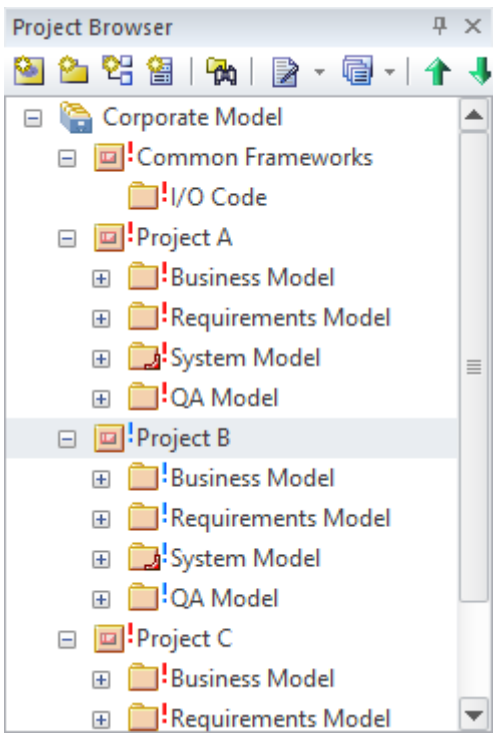
- User Security
- Visibility Levels
- Single Sign-On (SSO)
- Import User IDs From Active Directory
- Configure OpenID

Multiple Projects

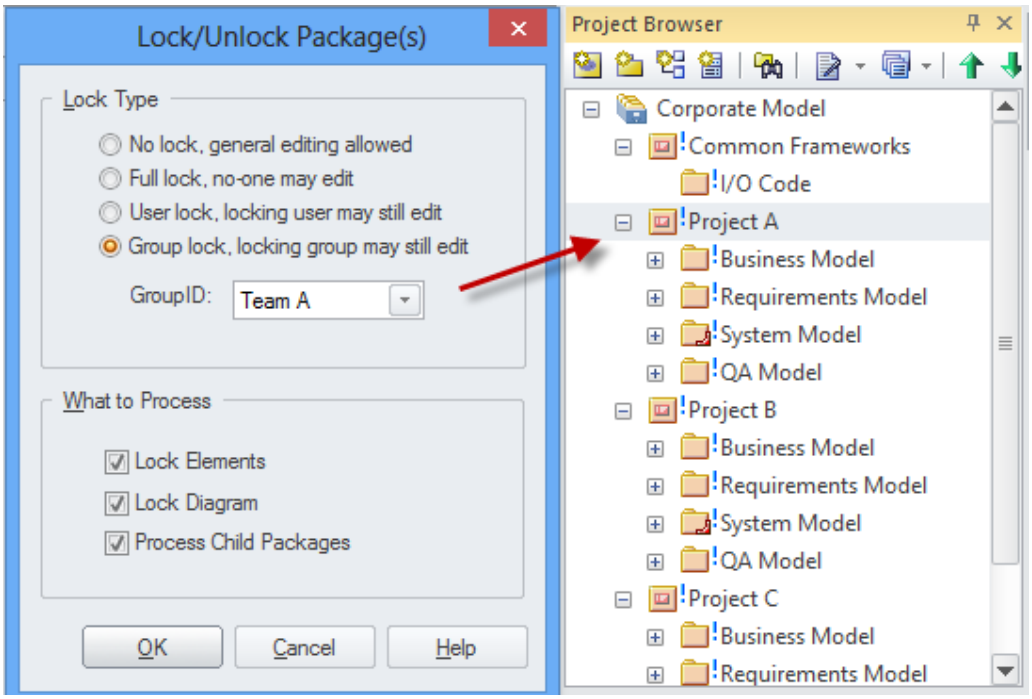
When developing multiple projects you might have common assets such as code foundation classes, scripts, templates and reports that are required by all the projects. As Enterprise Architect is only limited by the DBMS size, the cleanest method of supporting this is to combine all projects into one repository and set the security access rights to restrict specific groups of users to only make changes on their models. However, there can be project management scenarios that require multiple repositories with information interchange. The Multiple Project Options table identifies the options for these general scenarios:

- One Repository – Multiple Projects
- Multiple Repositories – Sharing Resources
- Large Projects – Branching and Merging Multiple Phases
- Global Sharing of Assets

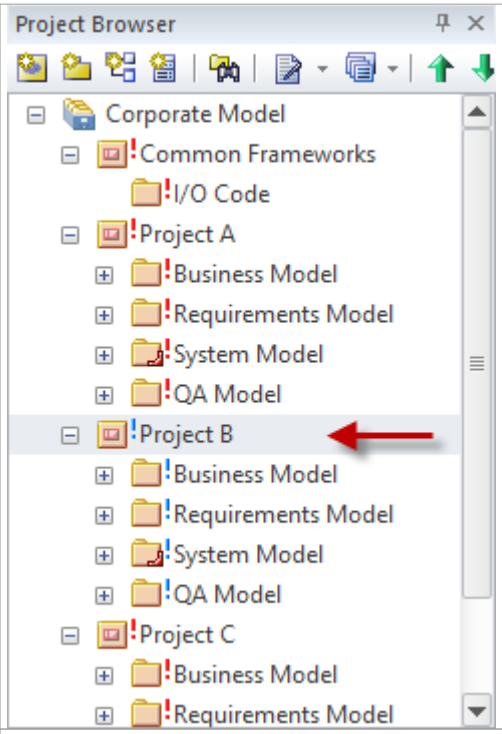
Multiple Project options

Option	Description
One Repository – Multiple Projects	Where you are using one model repository, it is best to set up the Browser window hierarchy with multiple project Root Nodes and multiple Views (see the image example). Each Root Node or View can represent a separate project. Commonality between projects (such as frameworks or foundation classes) can also be captured under a Root Node or View. Using a single repository will ensure that alterations to common code and references need only be made once to affect all projects.  Additionally, Enterprise Architect's security can be used to restrict the user's right to

make modifications in designated areas (Package sub-trees). Elements outside a designated area can be accessed for use in diagrams; however, data modification to these elements is restricted to the group of users that have security access to the Package-tree that these elements are contained in.
This example illustrates an administrator using group locking to restrict a Package View for a specific team to update.



The second figure shows the Browser window security locking when accessed by a user in the group 'Team B'. Note the coloring of the lock markers. The blue exclamation mark (!) indicates the locked Packages can be updated by the user currently logged in. The red exclamation mark (!) indicates the current user cannot update the corresponding Package contents.

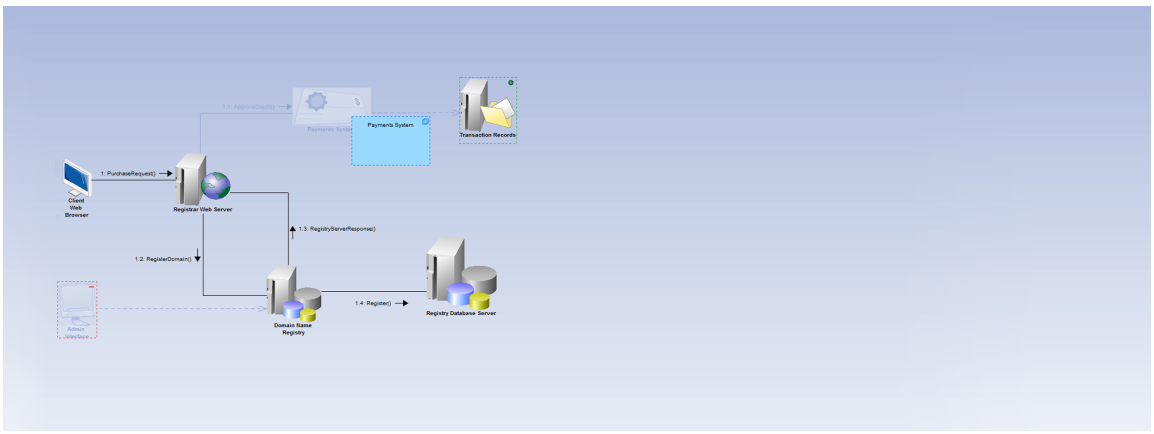
	
Large Projects – Branching and Merging Multiple Phases	In large 'phased' development projects, the design of the next phase can be carried out in a 'Branch' repository that is a copy of the 'Trunk' repository being used in the main development. After a phase of design work, 'Baselines' can be created against the Branch model and using the Baselines **Load Other Baseline** feature, the Branch can then be merged back into the Trunk model. With the combination of MDG Technologies and a Package transfer using the Baselines Merge 'Load Other Baselines' feature you can set up a number of repositories that interact with common data/resources and keep them up-to-date.
Multiple Repositories – Sharing resources	If you have multiple projects on multiple repositories, and you need to use the same resources across these projects, you can use a number of features for exchanging or accessing common resources. In all cases, it is recommended that a master repository is used to maintain a single source for the common project data. In this scenario there are two key sets of data: Packages (that is, modeling Packages and common frameworks) and Resource data (such as type definitions and report templates). There are several options for interchanging both of these sets of data between project repositories and a master repository. **Packages:** • Baseline Difference and Merge • RAS • Package Control • Version Control • XMI import/export **Resource data:** For resource data interchange there are three core options: • **Link Reference Data to a Shared Repository** A shared repository can be configured to be used for hosting a common set of system resources, including resources such as Security Users and Groups, Glossary terms, Data Types and Report Templates • **Reference Data Import/Export**

	Reference data can be periodically updated from the master repository using the **Export Reference Data** option • **MDG Technologies** Using MDG technologies you can build a module of resources that you can export; this MDG Technology can then be referenced by any Workstation instance of Enterprise Architect.
Global sharing of Assets	For scenarios where there are multiple repositories scattered across different organizations, the Reusable Asset Service (RAS) can be used to collaboratively exchange model data. A good example is when collaborating with contributors working for a number of different corporations, on modeling an industry standard. This requires a global access point with options to contribute new modeling or download the latest changes implemented to the specification/framework.

See also

- User Security
- Package Baselines
- Manage Baselines
- Change Management
- Link Reference Data to a Shared Repository
- Version Control
- Sharing Reference Data
- Reuseable Asset Service (RAS)

Change Control



Enterprise Architect supports a number of features for monitoring and controlling changes to the model. Each of these features has its different usage depending on the organization of your model and the content that is being developed. The features include:

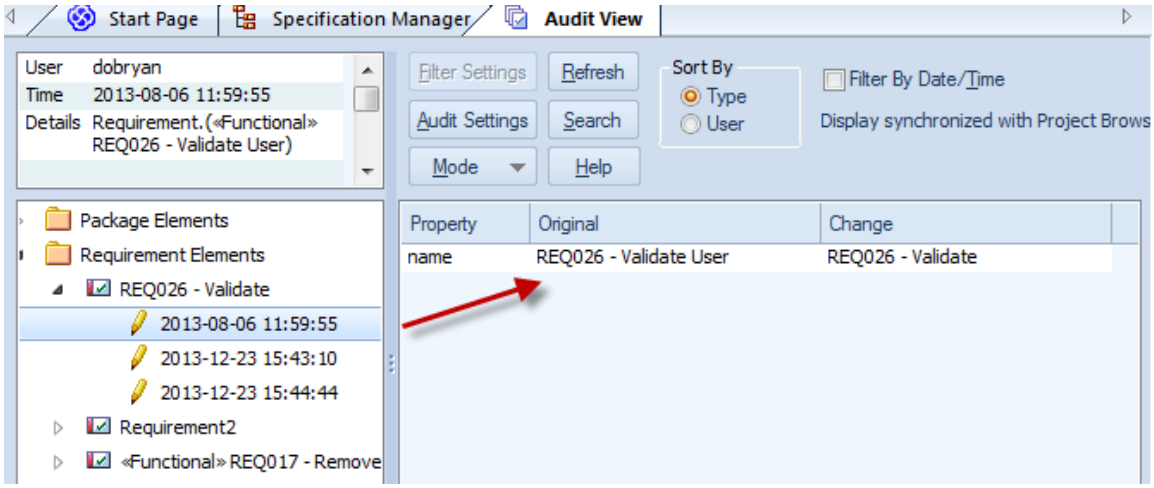
- **Auditing**
  Auditing provides continuous tracking and logging of changes in Enterprise Architect
- **Time Aware Modeling**
  Time Aware Modeling enables you to create incremental versions of your models, which you can use to easily compare 'As-Is' and 'To-Be' models, by analyzing a diagram to see exactly what changes occur at each iteration
- **Baselines**
  The Baseline Management feature provides a periodic means of tracking changes, along with support for comparing and merging changes
- **Version Control**
  Enterprise Architect supports Version Control of Packages and their sub-Packages via a central Version Control repository, which is maintained by a third-party Version Control application

These features can be used singularly or in combination; for instance, you might want to apply both Auditing and Baselines together. The benefits and shortfalls of each of these features are discussed in subsequent topics.

Auditing

The Audit feature enables you to record model changes in Enterprise Architect. It records details of who changed an element, when, what was changed, and the prior state of the model. This can be particularly useful for recording a history of changes to Requirements models.

This is an example of viewing alterations to an element directly in the Audit View. The illustration shows a number of alterations with the details of a selected change shown on the right.



With the Auditing View enabled the System Output window 'Audit History' tab shows the list of changes for the selected element.

The System Output window can be accessed using the ribbon: Start > Desktop > Design > System Output (Ctrl+Shift+8).

Although the Audit features provide a useful history of change, the data required over a long period or with large numbers of users can be taxing on the repository (DBMS). This can be avoided by keeping a fixed audit period and creating Baselines for each period, then saving and clearing the audit log. This can provide a longer term, lower volume, but more segmented history.

See also

- Auditing

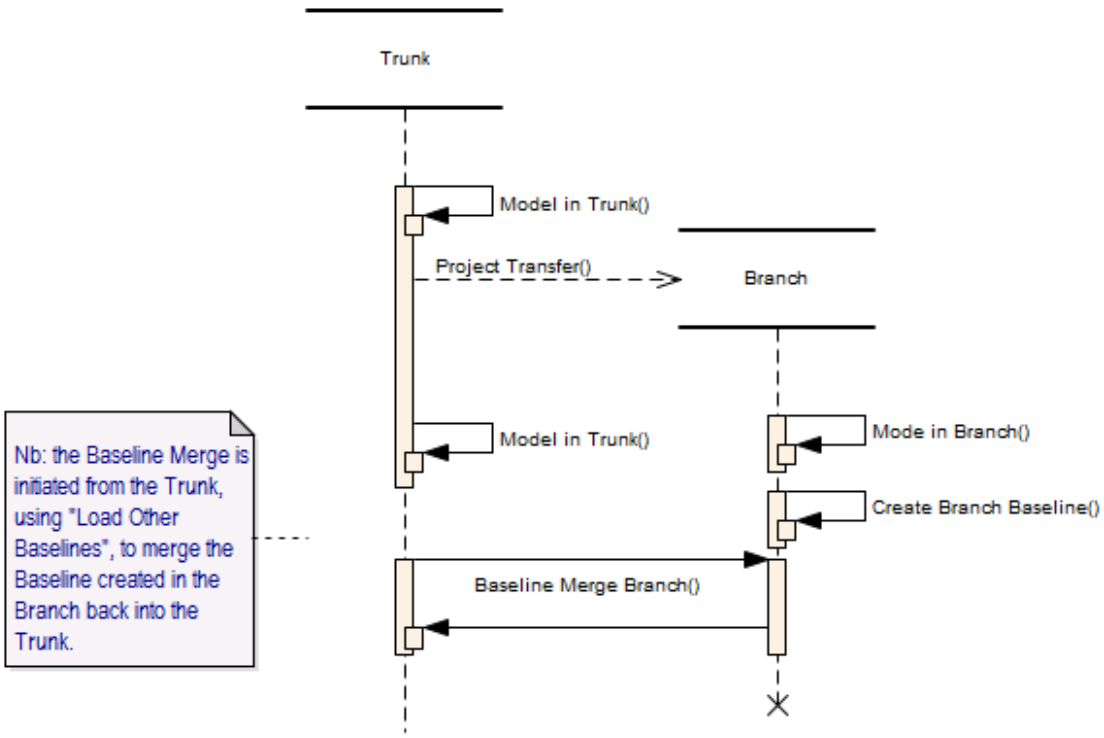
Baselines

The auditing feature provides continuous tracking and logging of changes to the model. The Baseline Management feature provides a more periodic means of tracking changes, along with support for comparing and merging changes. It allows Baselines of a model to be created on a periodic basis (such as by month, phase, version or build). Baselines can then be compared to the current model and changes selectively rolled back.

Branching using Baselines

Baselines can also be used for 'Branching' by creating a duplicate repository (a Branch) from the source repository (the Trunk). The branch can then be updated.

After updating the model in the Branch repository, you can merge the changes back to the Trunk repository by creating a Baseline in the Branch and then using the 'Load other Baselines' feature on the Trunk repository.



For more information on setting up baselines and viewing differences see the *Package Baseline* Help Topic

Package Baseline

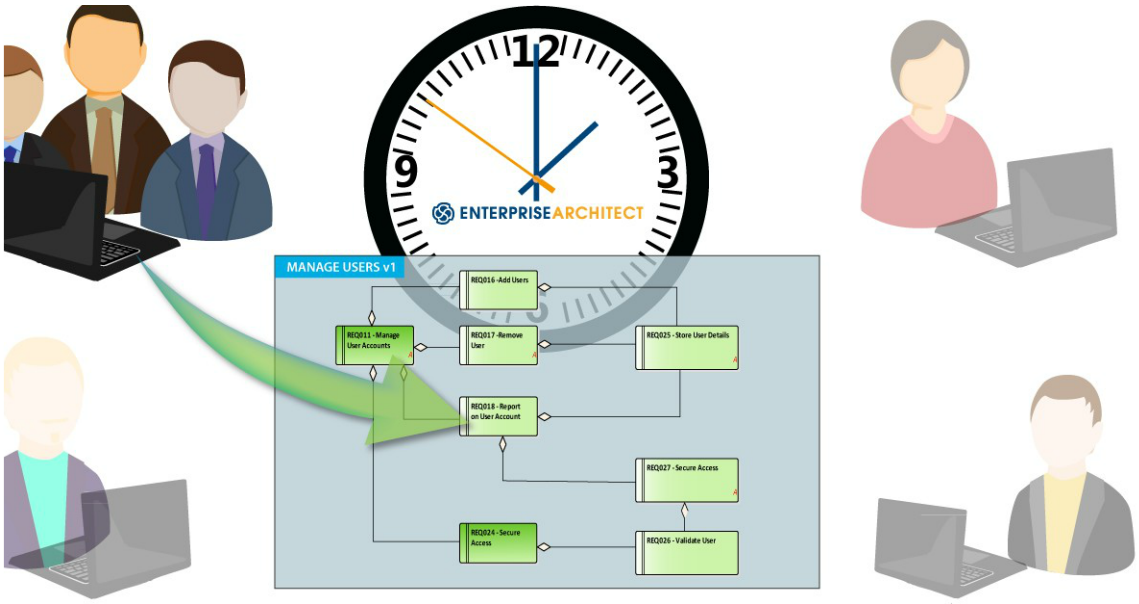
See also

- Package Baselines
- Manage Baselines

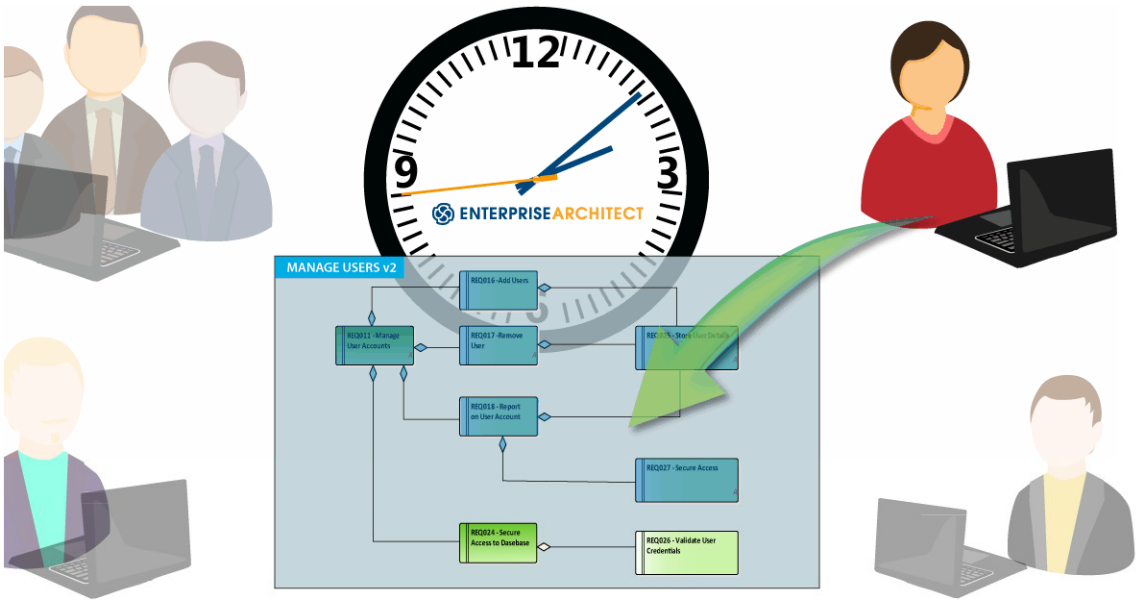
Time Aware Modeling

Using Time Aware Modeling you can create incremental versions of your models that allow you to follow their progression through various stages. It supports the migration of elements, diagrams and Packages through time as transitions or versions.

The process starts with a model seen as the current state or 'As Is' baseline.



Any potential future states are constructed from this providing for analysis or visualization of the 'To be' models.



Although Time Aware Modeling applies to many different analysts, architects and engineers alike, and many different scenarios and processes, a classic example for using it is where software is undergoing a version upgrade and the existing functions are being architecturally analyzed for potential new features. It shows how the existing entities in the model will change with the new features and what new entities are required to facilitate these features.

See Also

- Time Aware Modeling

Model Based Version Control

Enterprise Architect supports Version Control of Packages and their component sub-Packages to a central Version Control repository, which is maintained by a third-party Version Control application. This provides two key benefits:

- Saving a history of changes to Enterprise Architect Packages, including the ability to retrieve previous versions
- Coordinating the sharing of Packages between users

Version Control can be set up using any Version Control software that is compliant with these standards:

- SCC standards
- CVS
- Subversion
- Microsoft TFS.

Why use Version Control?

Although code-based Version Control is used extensively in code development, it is not so easy to implement model based Version Control. A core reason is that code is a simple text based file that does not support explicit cross referencing, making it far simpler to be Version Controlled than a model that has links referencing other parts of the model. The introduction of tightly controlled cross-referencing limits some of the features that are available in code based Version Control – it resembles trying to use a two dimensional tool in a three dimensional scenario.

This is not to say code-based Version Control should not be used in modeling, but rather to outline why it is recommended to explore the alternative options available in Enterprise Architect before exploring the more complex scenario of using a code-based Version Control system.

Benefits:

- More streamlined, where regular tracking of change is required
- A history of revisions can be viewed and restored
- A version packet can be a Package (not a full Package tree)
- Package Locking can be used to avoid editing conflicts

Shortfalls:

- The versioned data is stored external to the repository; this data, being separate to the modeling repository, can be lost, whereas Baselines are maintained within the model repository
- Due to the processing required to interface with the Version Control repository, the granularity of versioned Packages must be kept small; this creates complications with cross-referencing
- Version Control locking can restrict access to models, especially when Packages are left checked-out by a user
- For cross-continental usage the interchange with a Version Control repository can be very slow
- When using multiple Projects, only Package data is Version Controlled; reference data such as Document Templates, Calendar Events and Glossary must be manually exported/imported into each Project
- If users perform frequent imports from the Version Control system it can be detrimental to the

efficiency of other users, due to locking

- External Version Control can introduce configuration issues as these settings need to be configured by each individual user, which is a complicated process and if set incorrectly can result in data loss or corruption

Alternatives

The alternative options include using Auditing, Baselines, Pro Cloud Server and RAS, which can be used in different combinations to achieve a similar outcome to Version Control depending on what is required.

These points cover the broad features of Version Control, along with the alternatives that Enterprise Architect offers.

Model Sharing	**Cloud:** • Pro Cloud Server allow cross-site connectivity, avoiding the need for multiple repository interchanges • Reference data such as Document Templates, Calendar Events and Glossary do not have to be updated and interchanged. **RAS:** • RAS, using the Pro Cloud Server, supports cross-continent interchange of models as well as a version-based history of items. **Baselines:** • Baselines can be used for multiple repository interchange (including cross-site interchange) • Baselines allow for a more refined selection of alterations to be re-instated (merge feature) Baselines support Trunk and Branch cycling of repository data; Version Control for models does not.
Revisions	**Auditing:** • Auditing provides a simple log of alterations for viewing a history of changes. **Baselines:** • Baselines support periodic storage of models over a period • History of changes is available and differences can be viewed.
Package Locking	**Security:** • Where locking is required, rather than using Version Control use the Security lock in edit mode to ensure users are not attempting conflicting editing

Summary

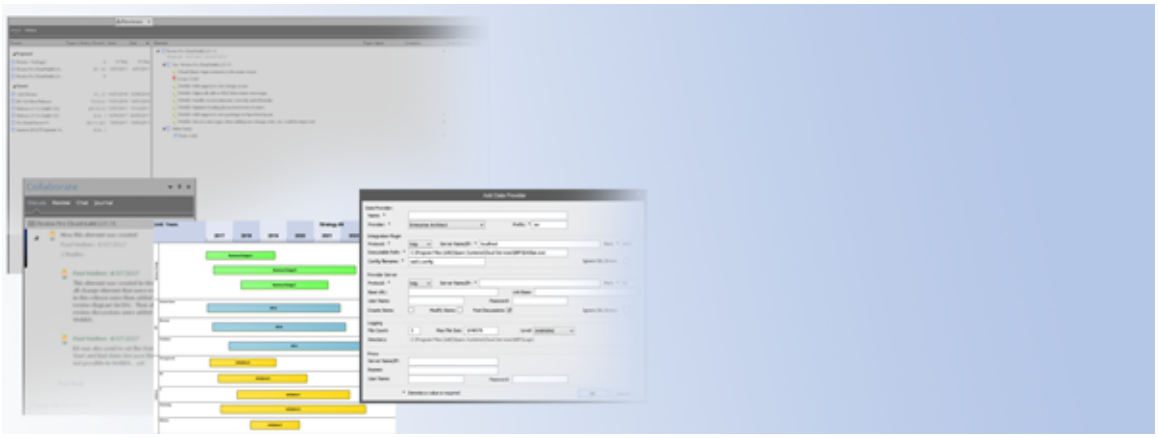
So, in short, where a history of changes is required it is suggested that you first weigh up the pros and cons of these alternatives before deploying Version Control.

Note: It is assumed that with Version Control you have a low user count on each repository and that this justifies using .eap files. However this is not recommended for large .eap files, as these can cause corruption in processing the XMI interchange for Version Control.

See Also

- Version Control
- Version Control White-paper.
- Auditing
- User Security
- Package Baselines
- Manage Baselines
- Change Management
- Reuseable Asset Service (RAS)

Team Collaboration

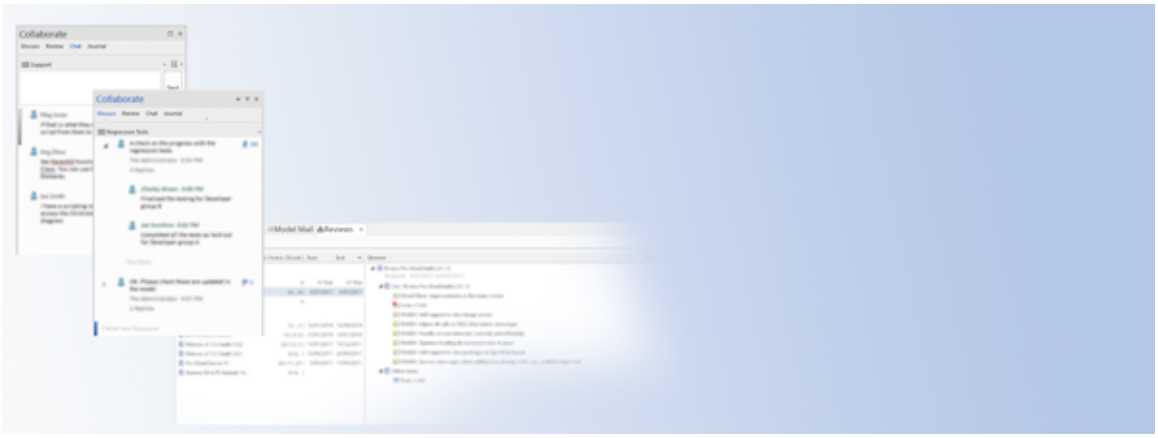


When working with inter-disciplinary teams it is crucial to have mechanisms for collaborating on both resources and ideas. Enterprise Architect supports a variety of features for team collaboration, including Discussion points, a repository-based forum, Model Mail and the means of sharing resources such as Profiles and report templates (MDG Technologies). When deploying Enterprise Architect for team work it is useful to consider how these features can be used.

The sections discuss:

- **Team Collaboration Features**
  The core Team Collaboration features include Model Mail, Discussions of elements, Reviews, Team Library
- **Team Collaboration Tools**
  The tools used to facilitate team collaboration include: Workflow Scripting, Security, MDG Technologies, Reference Data Import and Export

Team Collaboration Features



The core features for collaboration include interaction on a one-to-one basis using Model Mail and group based discussion on elements, through to a forum type interaction using the Team Library feature.

Discussions

Discussion points can be posted against any element or diagram. In a large team of developers it is essential that comments on an element can be directly posted for group review and reply, providing an interactive team-based process. Discussion differs from Model Mail in that it is a chronological sequence of comments on a specific element or diagram.

Where security is enabled on the model the identity of the Discussion's source is provided with the post.

Reviews

Reviews provide additional features to Discussions by supporting a more formal processing of a Discussion point lodged against an element or diagram. A Review allows the posted topic to be set with a Status and a Priority. It also provides a tabular summary of the status of the review's that are pending .

Model Chat

Model Chat allows you to select a user, or a group of users, and post a message for a two-way dialog or discussion. This facility provides a means of participating in quick conversations on a point of interest, with members of a selected user group.

Journal

The Journal provides a space for jotting down points like; what task have been completed, or to-do lists for keeping track of a user's personal processes.

Team Library

Enterprise Architect's Team Library facility helps users to discuss the development and progress of a project. Team members can view and post messages within the modeling environment and can link their posts directly to elements within the model. For distributed team environments, users can connect their Enterprise Architect model to a Team Library hosted in a remote model repository.

See Also

- Model Mail
- Discussions
- Formal Reviews
- Team Library

Team Collaboration Tools

The tools supporting Team Collaboration include workflow and security features that identity the user and assign tasks, features for data sharing across repositories, as well as features for interchanging data with external applications.

Workflow

In a model driven development environment there can be many workflow processes operating in the design and the development of a project.

Using Enterprise Architect's Workflow scripting you can set the order of work to be performed by members of the team and ensure that any specified outcomes are obtained on completion of the Workflow routine.

Workflow scripting is intended to be used by those administering the overall project management of a design and development process. As an administrator, you can use workflow scripting to define your own solution to a workflow process.

User Identity - Security

There are numerous benefits in enabling Security. These range from providing simple user access rights, through to providing details for many of the multi-user features of Enterprise Architect.

Some of the key team based facilities that use Security include:

- Workflow scripting
- Project Management Gantt charts and Calendar
- Model mail
- Audit tracking
- Discussion
- Reviews

Integration

In a model driven development environment there can be external applications used as in the development process. When using the Pro Cloud Server you can integrate data from external providers into an Enterprise Architect Cloud model. A variety of third-party providers can be connected to including ALM, Jazz, Jira and ServiceNow. Integration supports browsing the external provider's items and retrieving lists of elements and

objects based on the provider's queries. Capabilities include:

- Linking an Enterprise Architect element to an external object
- Viewing external element properties
- Import and Export elements
- View and in some cases add to, external object discussions
- Open external tools in a web browser at relevant bookmarks

MDG Technologies

In large organizations it is common to have different teams developing different models but needing to share common resources. These resources can range from company specific language Profiles used in modeling, through to coding templates and company reporting templates.

With an MDG Technology you can set up common resources to be used by groups using different repositories, across your organization. These shared resources include:

- Profiles (for defining or modifying a modeling language)
- Patterns (for creating model structures for re-use)
- Tagged Value Types (for setting user-defined fields)
- Code Modules (Code generation templates)
- MDA Transforms
- Report Templates
- Linked Document Templates
- Images
- Scripts
- Workspace Layouts
- Model Views
- Model Searches
- Model Templates

Once created, an MDG Technology can be deployed on a common network drive and referenced by settings within Enterprise Architect.

Reference Data Import/Export

As a simpler alternative to using MDG Technologies, the Reference Data Import/Export feature can be used for supporting multiple projects that use common data. Typically a master repository is kept up-to-date and a selection of the reference data is periodically propagated to other repositories.

Parts of a repository that are shared include:

- Glossaries
- Type definitions (such as status types)
- Resources, Clients
- RTF and HTML templates
- Security

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

See Also

- Workflow Scripts
- Workflow white paper
- User Security
- VIsibility Levels
- MDG Technologies
- Integration Plug-ins
- Sharing Reference Data

Optimization

When sharing large models in a large user group over a network (LAN, WAN or a Cloud), there are core considerations for optimizing the performance. In general, the performance depends on:

1. The model repository type used
2. The network response time.
3. The protocol used for the workstation connection to the repository.

When making decisions about deploying your repository there are two main areas to consider:

- Repository Optimization
- Network Optimization.

We will now discuss in more detail the options to consider when optimizing performance.

Repository Optimization

When optimizing the repository performance for high volume usage, the factors to consider are broad because:

- Each DBMS has different response times for different features
- Each organization can have vastly different usage of Enterprise Architect features, which means different emphasis on the usage of the tables in the repository
- Some organizations are tied to a specific DBMS brand, while others are not.

Given these factors, we can only recommend that you use your specific data-set, for comparing the various DBMS products that you have access to, and decide on what product you consider is of optimal performance. Then, based on the selected DBMS, you can consider the set up of the connection, the backup, the caching and the indexing.

Repository optimization options

Option	Description
Repository selection	Unless you are working on a small model that is personal or being shared over a group of less than five users, the repository is best housed in a DBMS rather than a file repository. EAP files are not considered reliable for high volume usage. Firebird file based repositories can be used for multi-user access, but only via a Cloud connection. Note that where tight security is required and the Visibility Level feature is required (Row Level Security), then this does narrow the selection down to either MS SQL Server or Oracle.
DBMS performance testing	Between the 5 DBMS products supported, there is a wide range of response times due differences in the data volume used for the same queries. Before choosing which DBMS to use for your Enterprise Architect repository, we do suggest you try a comparison of response times for the set of DBMS products that you are interested in using. A simple test is to: • Create a copy of a repository on each DBMS being tested - for example, start by using the EAExample.eap file, which can be easily transferred using the Project Transfer feature

	• Then for each repository generate out a common report to HTML • Measure the response time for each DBMS. MySQL and MariaSQL are both free and can provide a benchmark for comparison with other commercial DBMS servers. We have found that MySQL/Maria are very efficient, particularly for handling variable sized text fields and Blobs, which are the datatypes for Notes and document/images respectively. A recording of the timing of the same processes, when run on the different DBMS, will provide a good overview of the performance of each of the DBMS products that you are considering using. Enterprise Architect's *Project Transfer* feature provides a clean and simple mechanism for copying a repository from one DBMS to another for a quick analysis. **Note:** • Ensure that when comparing DBMS products that each DBMS is either installed on the same machine or at least a similar machine located on the same LAN. • It is recommended that if working on Cloud based repositories, that there should be a comparison between a locally installed DBMS server and a Cloud based one (like AWS). For instance, check the timing of the same Enterprise Architect process with the same DBMS, (with both on the Cloud machine), versus the timing of the same configuration and process on a local equivalent setup on a LAN. The reasoning behind this is that the performance on the off-site deployment can be influenced by many factors like; the network latency, how much load is on the network or the individual server.
DBMS Caching	Ensure that your DBMS is properly cached or is using an SSD drive. Caching is a critical starter in optimizing performance of any DBMS and it is crucial that your DBA implement this.
DBMS backup	Each DBMS product provides their own specific backup process. It is necessary to ensure that your Network administrator or DBA have scheduled regular backups of the Enterprise Architect repositories being hosted on the DBMS server.
Networking of servers	For small systems the Pro Cloud Server is best housed on the same machine as the DBMS to provide optimal through-put. Where the DBMS needs to be on a machine separate to the Pro Cloud Server then ensure that the two servers are networked appropriately over a high speed local LAN. Preferably using a 1 gb or faster ethernet connection between the Pro Cloud Server machine and the DBMS machine. The fastest is the best. If clustering of multiple machines is required for the DBMS ensure they are networked appropriately over a high speed local LAN and that any Pro Cloud Server is deployed on the same server-based high-speed LAN.
Indexing	The indexing provided for each DBMS is based on a broad review of different repository content. As that content can be vastly different for each corporation we do suggest, if response optimization is required, that you work with your DBA to test and optimize your indexing to best suit your organization's repository content.
OLE DB ODBC	Ensure that the connection to the database is optimal, as per the guidelines laid out for that DBMS-type repository setup. Where there is a choice the OLE DB connection is optimal.

See also

- Project Data Transfer

- Visibility Levels
- Server Based Repositories
- DBMS Repository scripts

Network Optimization

When sharing large models in a large user group over a network (LAN, WAN or a Cloud), there are two core factors to consider for optimizing the performance:

- Network load
- WAN latency

Enterprise Architect offers two options that have different network loads:

- **HTTP/S**
  HTTP/S is compressed and optimized for data transfer and is more efficient than a direct ODBC connection to a DBMS server.
- **ODBC**
  A direct workstation ODBC connection to the DBMS repository has been used in the past, however this gives a higher data load on the network and less efficient response than a HTTP connection.

So, for network efficiency and for optimal response, it is best to connect via HTTP/S using the Pro Cloud server.

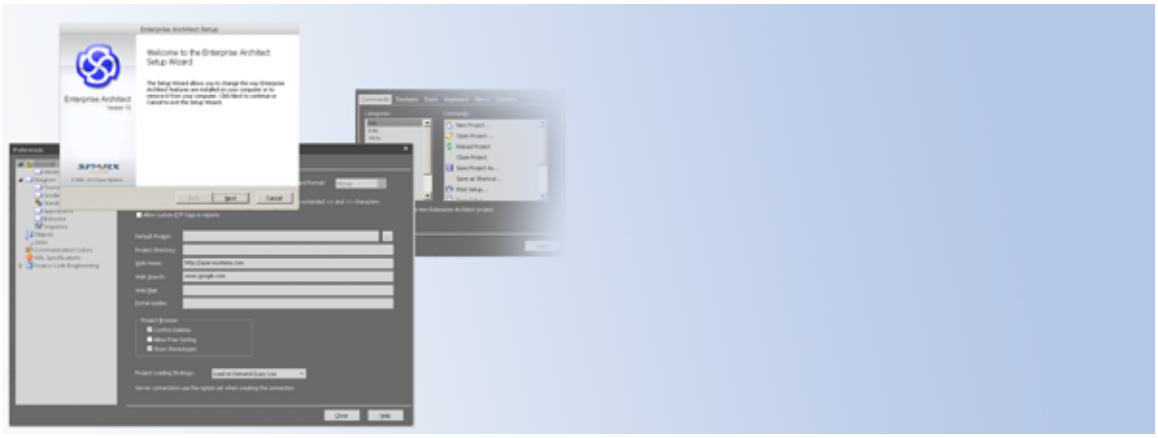
Network optimization options

When optimizing Enterprise Architect from an organizational networking perspective, consider these points:

Option	Description
DBMS and Cloud Server locations	When using the Sparx Systems Pro Cloud Server a core point to consider is where to position the Cloud and DBMS services. In general, for a small group system, it is optimal to have the Pro Cloud Server and the DBMS service running on the same machine. The Pro Cloud Server has minimal CPU and RAM usage. For very large groups of continuous users it might require separate servers for the Pro Cloud Server and for the DBMS. In this case it is critical to ensure that both the Pro Cloud Server and any DBMS servers are co-located in close proximity on the same high speed LAN.
WebEA clients numbers	The Pro Cloud Server's WebEA feature uses a very clean OSLC interchange that has a low data through-put and minimal query time. So, when comparing the load on the DBMS and the network traffic, for WebEA vs a workstation installation of Enterprise

	Architect, there is an order of magnitude difference. When using the Pro Cloud Server and an Apache server, for deploying WebEA: • For medium sized groups, this can be set up on the same server as the DBMS and Pro Cloud Server • For very large groups you might need to deploy a separate machine for the DBMS, the Pro Cloud server and the Apache server. To optimize this, these servers do need to be on the same direct LAN connection, however the performance will not be as good as having the PCS and DBMS components on the same machine.
Network latency DBMS ODBC	If using a workstation ODBC connection, in order to provide a reasonable response when modeling, the latency from a workstation to the DBMS server is best when it is less than 1-2 ms. Users find general performance less than ideal when network latency approaches 5ms. If the Latency is any higher you do need to consider using the Pro Cloud Server. The simple method to check this is to use a Ping command to check the time taken for a workstation to ping the IP address of the DBMS Server . For example, in a Windows Command prompt, ping to the server address and check the response: **ping** 192.168.0.10 /t If the response is over 1-2 ms and the repository response is slow, then it is best to consider using the Cloud service's more efficient HTTP connection.

Set up



This section provides an overview of the different set-up options available, ranging from a simple workstation install through to remote installations.

The common questions addressed in this section are:

- Do you want to install on each Workstation?
- Do you want to install on your Workstations remotely?
- Do you want a common fixed install?
- What repository type do you want to use?
- Do you want to use the Cloud service or ODBC direct?

The section discusses these installation options:

- **Workstation install**
  There are a number of options for workstation installation of Enterprise Architect; we discuss the standard simple install using the MSI installer
- **Remote Installation**
  When deploying Enterprise Architect across a network of workstations there are a number of applications and methods that can be used; we discuss the core options for remote installation
- **Simple Pro Cloud Server set up**
  We describe the initial set up of the Pro Cloud Server, using a file repository that can be used to test the service; once operating, it can then be configured to work with more detailed features such as a DBMS repository

Workstation Install

The standard workstation install simply requires that the Enterprise Architect installer file (MSI) is run on the workstation. The following process is for a standard Windows installation, however there is also comprehensive support for both Linux and macOS installations.

Install Process

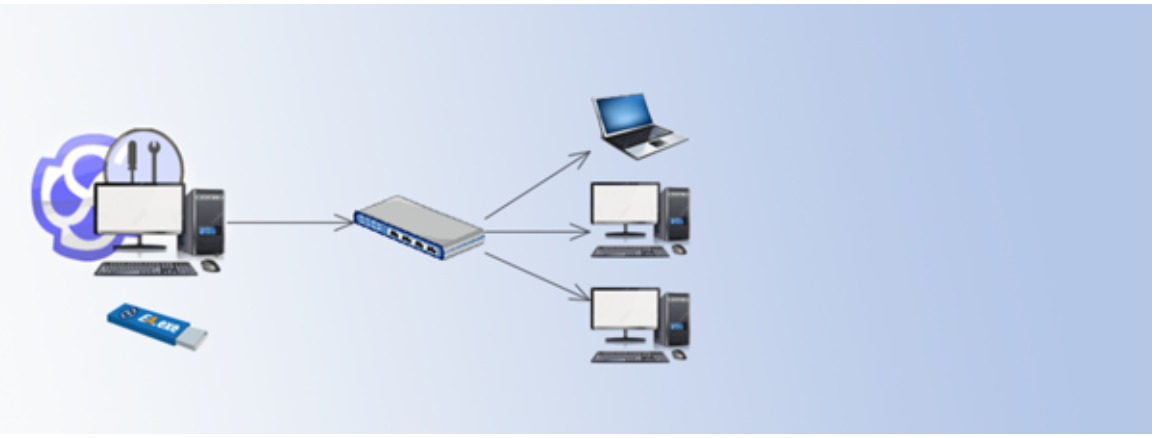
Step	Description
Check the System Requirements	The operating environment for Enterprise Architect is described on the System Requirements web page.
Download the Enterprise Architect installer	Download the MSI installer from the secure downloads site.
Install	To run the installer: • Double-click on the installation MSI file. The Enterprise Architect Installation Wizard screen displays. • Click on the Next button.
Licensing Agreement	Read the Licensing Agreement and, if you accept: • Click on the **I accept the license agreement** radio button. • Click on the **Next** button.
Readme	Read the Readme information. Click on the **Next** button.
Default folder setting	If you need to change the default folder for the installation, click on **Change** button. Then browse to the file path to install Enterprise Architect, then click the **Ok** and **Next** buttons.
Installation process	To begin the installation click on **Install** button. If a **User Access Control** dialog pops up, click on **Yes**. When the installation is complete, click on the **Finish** button.
Start Enterprise Architect	Start Enterprise Architect. The License Management dialog automatically displays.

Fixed License	If you are installing a Fixed license version you will need to add a License Key: • In the **Enter Private Key** option copy your Registration Key • Click on **OK** • Click on **Close** in the **License Manager** dialog.
Floating License	If you are using Floating Licenses you need to set up the Floating License Server and follow the Floating License Server scenario.

See also

- System Requirements
- Floating License Server
- Installing Enterprise Architect Under Linux or macOS

Remote Installation



When deploying Enterprise Architect across a network of workstations there are a number of applications and methods you can use. Deployment can be performed remotely using:

- Windows software deployment such as Microsoft SCCM
- Virtualized environments such as Citrix and other remote desktop clients
- Enterprise Architect's Zero Config Client

For each of these deployment options you use the Floating license Server or Keystore service to provide License keys, and you set up specific Windows registry keys to cover the options being deployed.

These topics describe the general set up process for these types of deployment:

- **Floating License remote install**
  This covers setting up a remote Enterprise Architect installation that references a Floating License Server
- **Remote install**
  How to use the Enterprise Architect MSI installer do the installation through Windows Server or SMS
- **Zero Config Client**
  How Enterprise Architect can be used on client machines without having to be installed on those machines

Note - for virtual machines the core topic is the remote set up of the connection to the Floating License Server.

Remote Install

An alternative option to the standard single workstation installation is to use a remote installation service for automating the installation of the application across the organization.

Remote Install Procedures

Remote Install	Enterprise Architect is set up using an MSI installer. You can use this file to do the installation through Windows Server or SMS. For documentation on the Command-line options available for Windows MSI installers see: https://msdn.microsoft.com/en-us/library/windows/desktop/aa367988(v=vs.85).aspx Note: By default Enterprise Architect will install for the Current User. To install for All Users specify: msiexec /i c:\easetupfull.msi /q allusers=2 A remote installation requires using the Floating Licensing service. For details on remotely installing the Floating Licensing see the *Remote Floating License* section.
Checks Post Installation	Please note that in this example Enterprise Architect was installed for All Users. 1. After installation check Windows | Add/Remove Programs for evidence that Enterprise Architect installed successfully. 2. Check the 'All Users' profile for desktop and start menu items to verify that Enterprise Architect was installed successfully for all users. Based on the operating systems see these directories: Windows 7+: C:\Users\Public\Desktop

Remote Floating License Installation

During automated installation of Enterprise Architect, registry entries can be set for each user, giving them access to a floating license key when they start Enterprise Architect. The registry settings differ for the file-based and the service-based keystores, as described here.

Note: it is recommended that the keystore service or Pro CLoud Server's Floating License Server be installed and set to your specific requirements. Then a workstation be set up to use this.

Once you are satisfied with that workstation's configuration; the following Registry entries can then be copied and propagated out to those machines that will be using the same configuration.

The following registry references are held under:

[HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS]

1. Example registry settings for the file-based keystore:

"SKT"=dword:00000000

"SharedKeyFolder"="Y:\\Dev\\Licenses"

"AutoCheckoutEx"=hex:1a,00,00,00

2. Example registry settings for the service-based keystore:

"SKT"=dword:00000001

"SSKSAddress"="ssks://pathToKeystoreService"

"SSKSPassword"="service password (encrypted)"

"AutoCheckoutEx"=hex:1a,00,00,00

3. Example registry settings for the Pro Cloud - Floating License Server:

"SKT"=dword:00000001

"SSKSAddress"="https://default@localcloud:443?user=default"

"SSKSPassword"=""Z?@k$wvaxzm2Ak"

"AutoCheckoutEx"=hex:1a,00,00,00

"SSKSSecurityFlags"=dword:00000000

Key Definitions

This table provides more detail on registry entries that relate to the KeyStore settings.

Setting	Description
SKT (Shared KeystoreType)	Specifies the type of keystore to obtain shared keys from. Permitted values are 0x00 (File based keystore) or 0x01 (TCP/IP based Service or HTTPS server based keystores).

SharedKeyFolder	This value should point to the mapped directory path, or network path, containing the shared sskeys.dat file. This setting is only used if the SKT key has a value of 0x00 (file based keystore). The example points to a directory on a network drive: Y:\dev\licenses. **Note:** • A full UNC path is recommended, for example: "\\DevelopmentServer\EA Licenses" • Enterprise Architect users must have read and write access to this file If accessing a key-file on a Novell server the file path is case-sensitive.
SSKSAddress	The ssks address to the Shared Keystore Service endpoint. This setting is only used if the SKT key has a value of 0x01 (service based keystore or HTTPS Server based Keystore). This is best copied from an example workstation that is setup with the Keystore or Pro Cloud Floating License server.
SSKSPassword	If the shared keystore service requires a password, it can be entered into this value. Note that this value is encrypted and cannot be entered in plain text. This setting is only used if the SKT key has a value of 0x01 (service or server based keystore)
SSKSSecurityFlags	This specifies the warnings relating to the PCS server certificate that should be suppressed when connecting to PCS for a license.
AutoCheckoutEx	Indicates which product keys Enterprise Architect should automatically try to obtain on start-up. Each key is represented by 4 bytes; for example: 1a 00 00 00 Where bytes 1-2 are the license code (1a00) and bytes 3-4 are the license type flag (0000). The allowable values are listed here.

License codes for AutoCheckoutEx

License types for AutoCheckoutEx:

Full License: 0000

Academic License: 0100

License types for AutoCheckoutEx

0200	Enterprise Architect Corporate
1a00	Enterprise Architect Ultimate
1800	Enterprise Architect Business & Software Engineering
1900	Enterprise Architect Systems Engineering

0a00	MDG Integration for Visual Studio
1400	MDG Integration for Eclipse
0300	MDG Link for Visual Studio
0800	MDG Link for Eclipse
0e00	MDG Link for Doors
1000	SysML
1200	MDG Technology for DDS
1600	Zachman Framework
1d00	TOGAF
1b00	MDG Technology for UPDM
0c00	RAQuest

Zero Config

For scenarios where a quick and simple deployment of Enterprise Architect is required, rather than having to perform a full install for each user, you can set it up in a simple read-only shared folder on a network drive or even on a USB drive.

Scenarios where this is a very useful solution include:

- Providing an online access for users to quickly start Enterprise Architect
- Setting up a Virtual Machine to host Enterprise Architect for brief use in an organization
- For contractors in client organizations where the client requires that they use their on-site machines

Advantages

- Keep Enterprise Architect up to date without the need to roll out updates on new versions
- Keep everyone on the same version
- Works well with Floating Licenses

Shortfalls

- There are restrictions to certain features with this method; for more details see the Help topic *Zero Config Client Support*

See also

- Zero Config Client Support

Pro Cloud Server install (unlicensed)

The Pro Cloud Server is an HTTP connectivity service that can be deployed on your internal LAN based server or on an external web based server. It can also be deployed on hosting services such as AWS or Azure.

An unlicensed installation of the Pro Cloud Server provides basic Cloud Repository functionality, allowing you to host Cloud repositories and access them via Enterprise Architect. With a valid license, the Pro Cloud Server provides a range of additional features.

Simple Set up

With the numerous options available for the Pro Cloud server it is recommended that the initial set up is kept simple by using purely the default settings. This provides a foundation that you can then use for adding other features.

With this simple set up you will run through:

- Installing the Pro Cloud Server
- Creating a simple database (without ODBC connections)
- Connecting from Enterprise Architect to the Pro Cloud Server
- Opening a Cloud repository from a workstation

Requirements

For this exercise you will use a Windows machine that has Enterprise Architect installed on it. For testing this you will use a second machine with Enterprise Architect installed.

Pro Cloud installation

Step	Description
Cloud Server Install	To create the Cloud service simply run the installer as outlined in the *Pro Cloud Server Installation* help topic. Leave the configuration file with the defaults supplied.
Configure Firewall	By default Windows has a firewall set. Check that the firewall on the machine has the **In Bound** rules set to accommodate the ports. See the **Firewall Settings** section in the *Pro Cloud Server Configuration* Help topic.
Open the Pro Cloud Server Configuration Client	Start the Pro Cloud Server Configuration Client. By default this is accessible from: ..\Program Files (x86)\Sparx Systems\Pro Cloud Server\Client\SSProCloudClient.exe
Connect to the Pro Cloud Server	In the **Server Address** use: localhost:803. Enter the default password: pcsadm1n Click on **OK**. This will open the Sparx Systems Pro Cloud Server Configuration Client dialog. Note, if you are using PCS version 4 or earlier, the default password is blank (leave the

	password field empty).
Create a Firebird Database	To set up a single Firebird DBMS: • Click on the **Add** button. The **Add Database Manager** dialog displays. • Type in a name for your Firebird database, followed by a .feap suffix. For example: **MyFirebird.feap** • Click on **OK** to save this.
Configure the Database	To enable access to your Firebird database: • From the **Database Managers** list, select the new database (**MyFirebird**) • Click on the **Edit** button to open the **Configure Database Manager** window. • Tick the **Enabled** option. If you are using the Pro Cloud Server and intend to access this model with WebEA, then also enable the **Enable Pro Features (OSLC, WebEA and Integration)** option. • Click on the **OK** button. • Click on the **Close** button.

Test the repository

Start Enterprise Architect, using either a shortcut or the EA.exe

Step	Description
Start Enterprise Architect	To run a simple test on the Cloud Connection:
Open a repository	Select  **| Open Project** from the main menu, this will open the 'Manage Projects' dialog
Select Connect to Cloud	Click on the **Connect to Cloud** button in the 'Manage Projects' dialog
Connect to the Cloud	Enter these values - with the Server being set to the IP address of the Server where the Cloud service is running (for example: 192.168.10): **Name:** My Firebird **Protocol:** http:// **Port:** 804 **Server:** localhost **Model Name:** MyFirebird Click on the **OK** button. This will open the new MyFirebird.feap repository ready for you to add your own Packages.

See Also

- Introducing Cloud Repositories - see the links under the *Webinar* heading.
- Pro Cloud Server Installation
- Pro Cloud Server Configuration
- Firewall Settings
- Pro Cloud Server Configuration Clients
- Add Database Manager
- Managing Connections to Projects

- Connecting Enterprise Architect to a Pro Cloud Server