



Getting Started With Enterprise Architect

Enterprise Architect is an intuitive, flexible and powerful UML analysis and design tool for building robust and maintainable software.

This guide provides an overview of Enterprise Architect, how to set it up and how to immediately start using it.



Copyright © 1998-2010 Sparx Systems Pty Ltd

Getting Started With Enterprise Architect

© 1998-2010 Sparx Systems Pty Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2010

Publisher

Sparx Systems

Managing Editor

Geoffrey Sparks

Technical Editors

Simon McNeilly

Bill Mioch

Brad Maxwell

John Redfern

Special thanks to:

All the people who have contributed suggestions, examples, bug reports and assistance in the development of Enterprise Architect. The task of developing and maintaining this tool has been greatly enhanced by their contribution.

Table of Contents

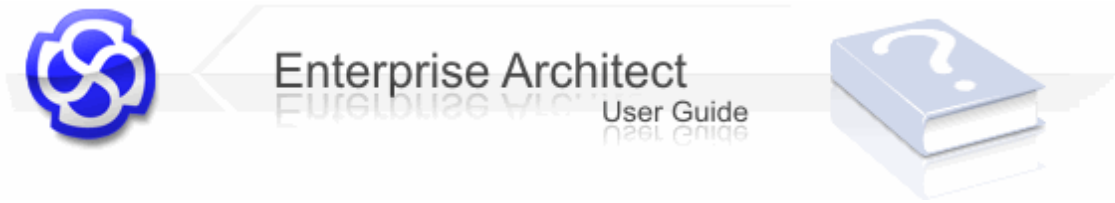
Foreword	1
Enterprise Architect Modeling Tool	2
What is Enterprise Architect?	4
Uses of Enterprise Architect	4
Key Features	7
Work with Enterprise Architect	8
Modeling.....	9
Manage Models.....	10
Code Engineering.....	11
Enterprise Architect Editions	12
Enterprise Architect Lite	14
Sparx Systems MDG Add-Ins	15
Help and Support	16
Available Helpfile Formats	17
Support	17
Formal Statements	18
Copyright Notice	18
End User License Agreement	18
Trademarks	21
Acknowledgements	22
If You Have the Trial Version	23
Order Enterprise Architect	25
Installation	25
Register a Full License	26
Start Modeling	28
Quick Start - Create a Project	29
Add a View to a Model	30
Add a Package To a Model	31
Add a Diagram to a Package	31
Add Elements	31
Add Connectors	32
Define Properties	33
Move Components	35
Delete Components	36
Save Changes	37
Quick Start - Project Tasks	38
Project Roles	40
Business Analysts	41
Software Architects	43
Software Engineers	45
Developers	46
Project Managers	48
Testers	49
Implementation Manager	50
Technology Developers	52

Database Developers	53
License Management	54
Finding Your License Information	55
Add License Key	56
Keystore Troubleshooting	58
Upgrade an Existing License	59
Register Add-In	62
Index	63

Foreword

This guide provides an overview of Enterprise Architect, how to set it up and how to immediately start using it.

1 Enterprise Architect Modeling Tool



Welcome to Sparx Systems *Enterprise Architect*, a UML 2.1 based modeling tool for designing and constructing software systems, for business process modeling, and for generalized modeling purposes such as visualizing existing systems and processes.

This user guide provides tutorials, guidance and reference material to help you get started with Enterprise Architect. You can then go on to perform:

- Modeling With Enterprise Architect
- Extending the Scope of UML With Enterprise Architect
- Requirements Management
- RTF and HTML Document Creation (Reports)
- UML Model Management
- Model Auditing
- Model Baselining and Differencing
- Model User Security
- Model Version Control
- Project Management
- Code Engineering
- Visualization of Code Execution (Debugging)
- MDA Transformations

See [What is Enterprise Architect?](#)^[4] for further details of what Enterprise Architect can do and what you can use it for.

Enterprise Architect makes extensive use of UML, so we provide a *UML Dictionary* of diagrams, elements and connectors. Enterprise Architect also includes a Software Developers' Kit (*SDK for Enterprise Architect*) that enables you to extend and customize the tool.

To Use This Guide

If you are new to modeling and UML as well as Enterprise Architect, or otherwise want a rapid review of the process of modeling with Enterprise Architect, go to the [Start UML Modeling](#)^[28] tutorial. This is not just a theoretical description - the first things you do are start Enterprise Architect and immediately create a model project.

Enterprise Architect is very flexible and has lots of features. When working through *Start UML Modeling* you will see many links to more extensive descriptions of features, functions, tasks and procedures, in *Using Enterprise Architect - UML Modeling Tool*. You could read *Using Enterprise Architect - UML Modeling Tool* in conjunction with this guide to get more information on any topic.

Using Enterprise Architect - UML Modeling Tool is the first of the main references for working with Enterprise Architect. The documentation includes further Enterprise Architect guides for each of the subject areas listed at the start of this topic.

You should read the Sparx Systems [Formal Statements](#)^[18], including the Copyright Notice and our End User Licensing Agreement.

Your Feedback

Sparx Systems likes to stay in touch with what Enterprise Architect users require in order to accomplish their tasks efficiently and effectively. We value any suggestions, feedback and comments you might have regarding this product, documentation or install process.

You can access our online feedback pages at:

- www.sparxsystems.com/bug_report.htm and
- www.sparxsystems.com/feature_request.htm.

Alternatively, you can contact Sparx Systems by email at: support@sparxsystems.com.

1.1 What is Enterprise Architect?



Powerful UML Analysis and Design Tool

Enterprise Architect is a comprehensive UML analysis and design tool, covering all aspects of the software development cycle from requirements gathering, through analysis, model design, testing, change control and maintenance to implementation, with full traceability.

Enterprise Architect combines the power of the latest UML 2.1 specification with a high performance, intuitive interface, to bring advanced modeling to the whole development team. It is a multi-user, visual tool with a great feature set (see below), helping analysts, testers, project managers, quality control staff and deployment staff around the world to build and document robust, maintainable systems and processes.

The UML Modeling Tool of Choice, Globally

With over 200,000 licenses sold, Enterprise Architect has proven highly popular across a wide range of industries and is used by thousands of companies world-wide. From large, well-known, multi-national organizations to smaller independent companies and consultants, Enterprise Architect has become the UML modeling tool of choice for developers, consultants and analysts in over 130 countries.

Sparx Systems software is used in the development of many kinds of application and system in a wide range of industries, including: aerospace, banking, web development, engineering, finance, medicine, military, research, academia, transport, retail, utilities (such as gas and electricity) and electrical engineering. It is also used effectively for UML and enterprise architecture training in many prominent colleges, training companies and universities around the world.

Now see:

- [Uses of Enterprise Architect](#) ⁴
- [Enterprise Architect Key Features](#) ⁷

1.1.1 Uses of Enterprise Architect

Enterprise Architect is a powerful tool for specifying, documenting and building your software and business process projects.

Using **Enterprise Architect's support for UML** and its related standards, you can model new complex software and business systems, or visualize and maintain existing systems.

Design and Build Diverse Systems Using UML

UML 2.1 is an open standard that provides a rich language for describing, documenting and designing software, business and IT systems in general. Enterprise Architect enables you to leverage the full expressive power of UML 2.1 to model, design and build diverse systems in an open and well understood manner. You can generate code, database structures, documentation and metrics; transform models; or specify behavior and structure as the basis for contractual agreements.

For further information, see *The UML Dictionary*.

Model and Manage Complexity

Enterprise Architect helps individuals, groups and large organizations model and manage complex information. Often this relates to software development and IT systems design and deployment, but it can also relate to business analysis and business process modeling. Enterprise Architect integrates and connects a wide range of structural and behavioral information, helping to build a coherent and verifiable architectural model, either what-is or what-will-be. Tools to manage version control, track and compare differences, audit changes and enforce security, help control project development and enforce compliance to standards.

For further information, see *User Security in UML Models*, *Baseline UML Models*, *Version Control Within UML Models Using Enterprise Architect* and *Auditing UML Models*.

Structured Use Case Scenarios

Enterprise Architect's Structured Scenario editor enables you to develop structured Use Case Scenarios, to capture vital analysis information in the form of natural language descriptions. The editor helps you use this information to drive downstream development and maximize traceability across the development life-cycle. The editor also helps you to dynamically link scenario steps to associated model elements, such as domain elements, business rules and glossary terms. From structured scenarios, you can automatically generate test case descriptions, and Activity and other UML behavioral diagrams. You can even reverse engineer existing process diagrams into structured, textual specifications to produce documentation deliverables.

For further information, see *Modeling With Enterprise Architect - UML Modeling Tool*.

Share Models

Enterprise Architect enables you to share complete models or specific aspects of a model between members of a team, including (through the ['Lite', read-only](#) ^[14] edition) stakeholders who can study a model but not change or manage it. You can make the project .EAP file available on a shared network drive, or replicate the .EAP file for complex distributed development. Alternatively, you can develop the project in one of several shared DBMS repositories, such as Access 2007, SQL Server; My SQL; PostgreSQL; Oracle 9i, 10g or 11g; and Sybase ASA. You can import and export data as XML files to distribute and update frameworks and other package-based model structures. You control changes through the version control repository. Enterprise Architect provides a data transfer wizard that enables you to upsize or downsize the complete model for maximum flexibility, and it enables you to export and import reference data so that you do not have to recreate it for related projects.

For further information, see *UML Model Management*.

Model, Manage and Trace Requirements

Enterprise Architect enables you to capture requirements and use full traceability from base requirements to design, build, deployment and beyond. You can use impact analysis to trace from proposed changes to original requirements, and build the 'right' system.

For further information, see *Requirements Management* and *UML Model Management*.

Develop Personal Views and Extracts of the Model

Enterprise Architect enables you to develop any number of different views of your model, or parts of it, either for your personal use or for the use of your team. These Model Views are generated by reports, so they can be set up to always show the current status of the selected view. The facility also enables you to create Favorites folders of hyperlinks to frequently-used data structures.

For further information, see *Using Enterprise Architect - UML Modeling Tool*.

Track and Trace Model Structures

In even a small model, it can be difficult to locate specific packages, diagrams or elements, even if you apply a rigorous naming and structure policy. Enterprise Architect has a wealth of facilities that enable you to locate structures quickly and easily, through the [Model Search](#), [Element List](#), Auditing facility, [Traceability](#) window, [Relationship Matrix](#) and reports. The [Element](#) menu, [Diagram](#) menu and [Project Browser](#) context menus also enable you to locate elements in diagrams and in the [Project Browser](#), and you can store hyperlinks to important or commonly-used elements and diagrams in the [Model Views](#). Finally, having located one element you can import any related elements into a diagram in a single operation.

For further information, see *Using Enterprise Architect - UML Modeling Tool*.

Generate Documentation

Enterprise Architect provides powerful document generation and reporting tools with a full WYSIWYG template editor for RTF or HTML output. You can generate complex and detailed reports from Enterprise Architect with the information you require in the format your company or client demands.

For further information, see *Report Creation in UML Models*.

Generate and Reverse Engineer Source Code

Enterprise Architect supports generation and reverse engineering of source code for many popular languages, including C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python and PHP. With a built in 'syntax highlighting' source code editor, Enterprise Architect enables you to quickly navigate and explore your model source code in the same environment. Code generation templates enable you to customize the generated source code to your company specifications.

For further information, see *Code Engineering Using UML Models*.

Visualize, Inspect and Understand Complex Software

Software is complex and often hard to understand. You can use Enterprise Architect to reverse engineer code in a wide range of software development languages and database repository schema, to understand static structure. To complete the picture, use the unique built-in profiling and debugging tools to capture and visualize executing software at run-time. Create run-time instances of model elements and invoke methods using the built in Object Workbench.

You can also bring in complete frameworks from source code or Java .jar files - or even .Net binary assemblies! By importing frameworks and library code, you can maximize re-use and understanding of your existing investment.

For further information, see *Code Engineering Using UML Models* and *Visual Execution Analyzer in Enterprise Architect*.

Perform MDA Transformations

Model Driven Architecture (MDA) is an open standard designed to facilitate rapid application development in a platform independent manner. Models can be built at a high level of abstraction and, using MDA based tools, transformed into models and code targeting a specific platform or domain.

Enterprise Architect supports advanced MDA transformations using easily edited and developed transformation templates. With built-in transformations for DDL, C#, Java, EJB and XSD, you can quickly develop complex solutions from simple platform independent models (PIMs) targeted at platform specific models (PSMs). One PIM can be used to generate and synchronize multiple PSMs, providing a significant productivity boost.

For further information, see the *MDA Transformations User Guide*.

SOA (Service Oriented Architecture) Support

Enterprise Architect enables you to rapidly model and forward- and reverse-engineer two key W3C XML technologies: XML Schema (XSD) and Web Service Definition Language (WSDL).

XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the coupling of UML 2.1 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.

For further information, see *Code Engineering Using UML Models*.

Systems Engineering support

Integrating many high-end features for Systems Engineers, the [Ultimate](#)^[12] and [Systems Engineering](#)^[13] editions of Enterprise Architect provide built-in support for SysML 1.1, parametric model simulation, executable code generation, as well as model to code transformations for Hardware Description Languages and Ada 2005.

For further information, see *Extending UML With Enterprise Architect* and *Code Engineering Using UML Models*.

Model Databases

Enterprise Architect enables you to reverse engineer from many popular DBMS systems, including Oracle 9i, 10g or 11g; SQL Server; My SQL; Access 2007 and PostgreSQL. You can model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile, and forward generate DDL scripts to create target database structures.

For further information, see *Code Engineering Using UML Models*.

Customize Enterprise Architect

Enterprise Architect also includes a Software Developers' Kit (SDK) that enables experienced tool developers to customize and extend Enterprise Architect to suit the specific requirements of their organization with, for example, in-house UML Profiles, Add-Ins and Code Templates. The very detailed Automation Interface gives you access to most element features, major functions such as XMI import/export, and attached information. Most properties are fully writable from the automation client. The Automation Interface provides great support for plug-ins, with the ability to embed automation client windows in the main diagram view. The Interface is accessible from any automation-aware client language, such as VB, C#, C++ and Delphi.

For further information, see *SDK for Enterprise Architect*.

Link Enterprise Architect to IDEs

Using Sparx Systems Model Driven Generation (MDG) Link plug-ins, you can develop source code in your preferred Integrated Development Environment such as [Visual Studio .NET](#) or [Eclipse](#), use Enterprise Architect to locate the source code for Classes, attributes and operations, and to model, navigate, track, reverse engineer, build and run your project.

The MDG Integration products for [Eclipse](#) and [Visual Studio 2008](#) provide an even closer, seamless integration of Enterprise Architect and UML 2.1 with your IDE, bringing the functionality required of a fully fledged modeling platform right inside the IDE.

1.1.2 Key Features

Enterprise Architect is renowned for its rich feature set. Some of the key features are highlighted in the following list:

- Model complex information, software and hardware systems using UML-compliant notation (comprehensive **UML 2.1** support for all 13 UML diagrams)
- Extended modeling for **Requirements, User Interface Design, Mind Mapping, Data Modeling, SysML, SPEM, BPMN 1.1** and more
- Generate **BPEL** scripts automatically from **Business Process** models
- Built-in **Requirements Management** enables you to specify, trace and verify requirements directly against the design, right through to the deployed solution
- Comprehensive and flexible MS Word-compatible **HTML and RTF report options**
- Leverage industry-standard **Enterprise Architecture** frameworks (**Zachman, TOGAF, DoDAF-MODAF**)
- Support in **forward and reverse code engineering** for many software and hardware languages 'out of the box': ActionScript 3.0, Java, C#, C++, VB.Net, Delphi, Visual Basic, Python, PHP, Verilog, VHDL and SystemC
- Ability to perform **database modeling**, to **reverse engineer** from a range of DBMSs via ODBC, and to **forward generate DDL scripts** to create database structures
- Connect to **shared database repositories** using MS SQL Server, MySQL, Oracle and more
- **Manage, track and control change** using **baseline** model merge and **auditing** capabilities
- **Centralize enterprise-wide documentation** of processes and information systems
- **Model dependencies** between elements, system dynamics and state
- **Model class hierarchies**, deployment, components and implementation details
- **Record project issues, tasks** and system glossary
- **Assign resources** to model elements and **track effort expended** against required effort
- **Testing support** for test cases, JUnit and NUnit
- Integrated **Debug Workbench** for visualizing executable Java and .Net applications, instantiating run-time model objects and generating Sequence diagrams from a stack trace
- **Migrate changes** across a distributed environment using **Controlled XMI Packages**
- Manage **Version control** through XMI using **SCC CVS** and **Subversion** configurations
- Inbuilt user and group **security** and access control management
- **Distributed development** through shareable files, use of **shared repositories** in a range of major Database Management Systems, file replication, data transfer, and import and export of reference data
- **Share models** using the latest **XMI 2.1** format
- **Import models** in XMI format from other tools
- Built-in Model Driven Architecture (**MDA**) **Transformations**, and facilities to import or create others

- Facilities to **import database schema, XSD and WSDL source, .NET and Java binaries**
- Use **UML Profiles** to create custom extensions for domain-specific modeling
- Save and load complete diagrams as **UML Patterns**
- Create and share dynamic views of model elements and diagram sets using **Model Views**
- Analyze and trace relationships between elements using the tabular **Relationship Matrix**
- Generate **executable business logic** from **rule tasks** and trace to natural language **business rules**
- Transform **behavioral models** into executable source code for software and **hardware description languages** (HDLs) such as Verilog, VHDL, and SystemC
- **Simulate SysML parametric models**
- Script and automate common tasks using a detailed **Automation Interface** and **Model Scripts**
- A range of internal and external [commercial MDG Add-Ins](#)^[15] to integrate the facilities of Enterprise Architect with IDEs and other technologies, and templates to write your own
- **Read-only Viewer** enables stakeholders to view but not change milestone deliverables
- **Price:** Enterprise Architect is priced to outfit the entire team, making collaboration and team development a real possibility
- **Speed:** Enterprise Architect is quick to load and a spectacularly fast performer, even with large models
- **Scalability:** Enterprise Architect supports single users and the development of small models, or many concurrent users developing extremely large models, with equal ease
- **Usability:** many of our users agree, Enterprise Architect gets you started and productive quickly, with a rich user interface and the ability to create **templates, model views** and 'favorites' collections of commonly-used elements and diagrams.

For a complete list of the new features of the latest version of Enterprise Architect, click on the **Help | Read Me** menu option.

Enterprise Architect is available in six editions: **Ultimate, Business and Software Engineering, Systems Engineering, Corporate, Professional,** and **Desktop**, each of which offers a different range of features. For a comparison of the Enterprise Architect editions, see the [Enterprise Architect Editions](#)^[12] topic.

1.1.3 Work with Enterprise Architect



Enterprise Architect is a powerful CASE tool for specifying, documenting and building software projects. Using Enterprise Architect's support for UML and its related standards, you can model new complex software and business systems, or visualize and maintain existing systems.

This topic introduces the fundamental processes that Enterprise Architect supports.

Modeling with UML

Enterprise Architect is a comprehensive UML analysis and design tool. To create models with Enterprise Architect, you therefore should become familiar with:

- how Enterprise Architect **implements** the UML standards and
- how you apply UML in Enterprise Architect to **develop your models**.

For more information, see the [Modeling With UML](#)^[9] topic.

Managing UML Models

To manage the models in your projects, you both protect and manage the **model data** itself, and communicate information on the data in the form of RTF and HTML **documentation and reports**.

For more information, see the [UML Model Management](#)^[10] topic.

Code Engineering

In Enterprise Architect, UML modeling both depends on and supports code engineering - you generate and update code from a model, and you create and update models from code. In this broad sense, Enterprise Architect enables you to:

- **Forward engineer, reverse engineer**, round-trip and **synchronize** code in a **range of programming languages**
- **Debug and profile** code
- Model and generate code for **XML Technologies**
- Perform **database modeling** and database design for a **range of database management systems**
- Convert model components from one **domain** to another using **Model Driven Architecture (MDA) Transformations**.

For more information, see the [Code Engineering](#)^[11] topic in this section.

Managing Projects

Enterprise Architect provides strong support for Project Management, particularly in the following areas:

- **Project estimation** - working out how much time and effort is required to build and deploy a solution, using the **Use Case metrics** facility and carefully-calibrated **metrics**
- Defining, assigning and **managing resources**
- Monitoring and managing **problems, changes, issues and tasks** that affect both individual **elements** and the **project** as a whole
- Managing the development, execution and results of **testing**, from Integration through to User Acceptance, and
- Maintaining a **project glossary** of terms, procedures and policies applied to the project.

For more information, see *Project Management with Enterprise Architect*.

Project management discussions and decisions can be communicated to the project through the **Team Review** (see *Using Enterprise Architect – UML Modeling Tool*).

The scope of your project management might include upgrades to Enterprise Architect and installation of related technologies. In this case, also see [License Management](#)^[54].

Extending Enterprise Architect Facilities

Experienced Technology Developers can **develop customized additions** to the functionality already present within Enterprise Architect. These additions include:

- **UML Profiles** and **Stereotypes**
- **UML Patterns**
- **Code Templates**
- **Tagged Value Types**
- **MDG Technologies** and
- Enterprise Architect **Add-Ins**.

By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.

For more information, see *SDK For Enterprise Architect*.

1.1.3.1 Modeling

Enterprise Architect is a comprehensive UML analysis and design tool. It provides a library of UML data structures that you can use and extend to develop your models.

UML Structures

To explain how Enterprise Architect interprets the UML standards and specifications, Sparx Systems provides *The UML Dictionary* of diagrams, elements and connectors.

- You **create your projects and models** using the **Start Page** or **File Menu**, which provide templates on which to base your models

- You initially create your **packages** and **diagrams** using the Toolbars and Menus, and the **elements** and **connectors** using the Enterprise Architect UML **Toolbox**
- You can also create new structures through the **Project Browser**, and **re-use existing structures** using the **Project Browser**, **Model Views**, **Element List** and **Model Search**.

For further information, see *Using Enterprise Architect - UML Modeling Tool*.

UML Modeling With Enterprise Architect

Modeling in Enterprise Architect is the process of graphically representing a business process or software system. The resulting model can be used to emphasize a certain aspect of the system being represented and to record and communicate its detail.

Building models requires the use of various UML data structures and Enterprise Architect tools, as above. A further extremely useful tool is the:

- **Relationship Matrix**, which enables you to visualize and amend the relationships and hence organization of structures within the model.

Enterprise Architect also provides particular support for:

- **Requirements Management** and
- **Modeling the business process**, an essential part of any software development process.

You can extend the scope of your models by **using**:

- **UML Stereotypes, Profiles** and **Patterns**, and
- **MDG Technologies**.

For more information, see *UML Modeling With Enterprise Architect - UML Modeling Tool, Extending UML With Enterprise Architect* and *Requirements Management*.

1.1.3.2 Manage Models

To manage the UML models in your projects, you both protect and manage the model data itself, and communicate information on the data in the form of documentation and reports.

UML Model Management

In managing models, you control:

- The **model files** in a Microsoft JET database or (Corporate and extended editions) in one of a range of DBMS repositories
- Model **data integrity**
- Development of the models in a **shared**, team environment
- **Versions** of the model, ensuring that work on different areas of the model is coordinated and synchronous rather than conflicting
- **User security**
- **Transfer** of **value data** and **reference data** between projects and models
- Changes to model data, using model **auditing**, **Baselines** and a **differencing** utility that enables you to roll back changes to a previous state
- Model **upgrades**
- **Replication** of models for parallel development (.EAP repositories only)
- Extensions of development with **Add-Ins** and the Enterprise Architect **Automation Interface**

You can also have recorded discussion and communication of decisions using the **Team Review**.

For further information, see *UML Model Management*.

Generating Model Documentation

You can generate documentation from the components of your model, in RTF or HTML format. You can also generate a range of RTF reports on your model.

For more information, see *Report Creation in UML Models*.

1.1.3.3 Code Engineering

Code Engineering with Enterprise Architect broadly encompasses various processes for generating or transforming code from your UML model and importing code into the model, to support model development in several coding languages, database development and SOA development.

Code Engineering

Enterprise Architect supports:

- **Source code generation and reverse engineering** for many popular languages, including **C++**, **C#**, **Java**, **Delphi**, **VB.Net**, **Visual Basic**, **ActionScript**, **Python** and **PHP**.

Enterprise Architect also provides:

- A built in 'syntax highlighting' **source code editor**
- **Code generation templates**, which enable you to **customize** the generated source code to your company specifications.

For more information, see *Code Engineering Using UML Models*.

MDA Transformations

Enterprise Architect provides:

- Advanced Model Driven Architecture (**MDA**) transformations using **transformation templates**
- **Built-in transformations** for **DDL**, **C#**, **Java**, **EJB** and **XSD**.

One PIM can be used to generate and synchronize **multiple PSMs**, providing a **significant productivity boost**.

For more information, see the *MDA Transformations User Guide*.

Visual Execution Analysis

Enterprise Architect enables you to:

- **Build, test, debug, run and execute deployment scripts**
- **Integrate** UML development and modeling with source development and compilation
- **Generate NUnit** and **JUnit** test Classes from source Classes using **MDA Transformations**
- Integrate the **test process** directly into the Enterprise Architect IDE
- **Debug .NET**, **Java** and **Microsoft Native** (C, C++ and Visual Basic) applications.

For more information, see *Visual Execution Analyzer In Enterprise Architect*.

Database Modeling

Enterprise Architect enables you to:

- **Reverse engineer** from many popular **DBMSs**, including **SQL Server**, **My SQL**, **Access**, **PostgreSQL** and **Oracle 9i**, **10g** or **11g**
- **Model database tables, columns, keys, foreign keys and complex relationships** using UML and an **inbuilt data modeling profile**
- **Forward generate DDL scripts** to create target database structures.

For more information, see *Code Engineering Using UML Models*.

XML Technology Engineering

Enterprise Architect enables you to rapidly **model**, **forward engineer** and **reverse engineer** two key **W3C XML** technologies:

- **XML Schema (XSD)**
- **Web Service Definition Language (WSDL)**.

XSD and WSDL support is critical for the development of a complete **Service Oriented Architecture (SOA)**, and the coupling of UML 2.1 and XML provides the natural mechanism for implementing XML-based SOA artifacts within an organization.

For more information, see *Code Engineering Using UML Models*.

1.1.4 Enterprise Architect Editions

Enterprise Architect is available in six editions: the **Ultimate**, **Business and Software Engineering**, and **Systems Engineering** 'suite' editions, and the **Corporate**, **Professional**, and **Desktop** editions. Functionality for each edition is described below; the features of and differences between the editions are listed in the table provided on the [Sparx Systems website](#).

Tip:

To help you understand the differences between these editions and the advantages and limitations of each, the Trial version of Enterprise Architect can be opened in any required configuration. When Enterprise Architect starts, select the mode to trial; you can close down Enterprise Architect and restart it in another mode for comparison.

The fully functional 30 day trial version of Enterprise Architect is available free of charge at www.sparxsystems.com/bin/easetup.exe.



Ultimate Edition

The Ultimate edition is designed for power users and those working across multiple domains, providing deep support for Business, for Software Engineering and for Systems Development seamlessly integrated into a single development environment. It enables you to drill down to the lowest levels of systems design and construction, with SysML and executable code generation for standard and hardware description languages. Business users can leverage BPEL, the Rules Composer and executable UML, in addition to all the advanced features of the other editions of Enterprise Architect. Software developers can integrate their Eclipse and Visual Studio projects with their UML models and leverage the advanced executable code generators to target different domains.

This edition enables end to end traceability throughout a global vision of your enterprise - unifying strategy, business process, interfaces, software, rules, data and fine grained systems. Powerful tools, domain-specific technologies, frameworks, integration platforms and a consistent, scalable, and robust interface work in unison to help you deliver on the promise of Model Driven Development.

As explained above, the Ultimate edition incorporates a number of [MDG Technologies](#)¹⁵ and Add-Ins. The Ultimate edition and MDG Technologies are all available in either Fixed License or Floating License form. The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.

The Ultimate edition provides:

- Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VBNet
- Full round trip support for Hardware Description Languages (Verilog, VHDL and SystemC) including support for generating State Machine code
- SysML Simulation Support - including support for simulating SysML 1.1 constraint models with results graphing capabilities
- BPEL Generation – transform BPMN 1.1 Business process models down to BPEL 1.1 code
- Business Rules – trace from abstract business rules down to automatically generated behavioral code.



Business and Software Engineering Edition

The Business and Software Engineering edition is aimed at software development professionals, business

modelers, architects, requirements experts, project managers and others involved in the design and construction of quality software and business services. It combines powerful new features such as executable code generation from UML models, BPEL, advanced scripting and a multi-purpose Rules Composer targeting executable code from Business Domain models, and bundles licenses for integration products and frameworks such as DoDAF-MODAF, TOGAF and Zachman, to provide advanced model-driven construction tools to tightly bind your code development in Eclipse or Visual Studio.

As explained above, the Business and Software Engineering edition incorporates a number of [MDG Technologies](#) ^[15] and Add-Ins. The Business and Software Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form. The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.

The Business and Software Engineering edition provides:

- Generation of Behavioral Code from State, Sequence and Activity models, supporting standard programming languages such as Java and .NET
- Advanced math functions within the scripting engine
- BPEL Generation from BPMN 1.1 models - including validation and WSDL support
- A Business Rules Composer that enables you to build Business Domain models and generate code to implement complex business rules in standard programming languages.



Systems Engineering Edition

The Systems Engineering edition is designed for systems and software development professionals working on real-time, embedded and systems solutions. It combines new features such as executable code generation from UML models (including support for hardware languages such as Verilog and VHDL), Ada, SysML 1.1, executable SysML Parametric diagrams and advanced scripting, and bundles licenses for DoDAF-MODAF, SysML, DDS, TcSE and integration products to provide powerful model-driven construction tools for the Systems Engineering domain to tightly bind your code development in Eclipse or Visual Studio with the UML/SysML models developed in Enterprise Architect.

As explained above, the Systems Engineering edition incorporates a number of [MDG Technologies](#) ^[15] and Add-Ins. The Systems Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form. The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.

This edition provides:

- Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VBNet
- Full round trip support for Hardware Description Languages, including Verilog, VHDL and SystemC, with support for generating State Machine code
- SysML Simulation Support - Includes support for simulating SysML 1.1 constraint models with results graphing capabilities.



Corporate Edition

Aimed at larger development teams, the Corporate edition enables you to connect to the following DBMS back ends as the shared repository: MySQL, SQL Server, PostgreSQL, Sybase Adaptive Server Anywhere, Access 2007 and Oracle 9i, 10g or 11g. This provides additional scalability and improved concurrency over the shared .EAP file approach to model sharing. User security, user logins, user groups and user level locking of elements, user/group based security (with locking at diagram and element levels) are also supported. Security comes in two modes: in the first mode, all elements are considered 'writeable' until explicitly locked by a user

or group; in the second mode, all elements are considered locked until checked out with a user lock.

The Corporate edition forms the base for the three extended editions described above. Like those editions, it is available in either Fixed License or Floating License form. The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.



Professional Edition

Aimed at work groups and developers, the Professional edition supports shared projects through replication and shared network files. This edition has an ActiveX interface for interrogating Enterprise Architect projects and extracting information in XMI format. The Professional edition fully supports code import/export and synchronization of model elements with source code. It enables reverse engineering of SQL Server; MS Access 97, 2000 and 2003; and Oracle 9i, 10g or 11g databases. Support for MDG Technologies and MDG Link (sold separately) is included with the Professional version of Enterprise Architect. The shared repository available in the Professional edition is restricted to the .EAP file format (JET database).



Desktop Edition

The Desktop edition is targeted at single analysts and developers producing UML analysis and design models. It provides facilities for UML modeling, XMI import/export, document generation, version control integration and profile/metamodel extensibility.

1.1.5 Enterprise Architect Lite

Enterprise Architect Lite is a **free, read-only** edition of Enterprise Architect that enables people such as project sponsors to review the project without making any changes. However, users of Enterprise Architect Lite also have wider access to:

- The **Team Review**, where readers can create and respond to posts, and link their comments to elements (see *Using Enterprise Architect - UML Modeling Tool*)
- The **Source Code Viewer**, where readers can open and edit external source code files, debug code, and configure and run package build scripts (see *Using Enterprise Architect - UML Modeling Tool*)
- The **File** menu, where readers can copy the project or create a shortcut to access it (see *Using Enterprise Architect - UML Modeling Tool*)
- The **Relationship Matrix**, where readers can export the matrix contents to a CSV file to be opened in a spreadsheet application (see *UML Modeling with Enterprise Architect - UML Modeling Tool*)
- The **Default Hours** tab to review project metrics (see *Project Management with Enterprise Architect*).

You can download the Enterprise Architect Lite edition (as the *Enterprise Architect Viewer*) from the Sparx Systems website at <http://www.sparxsystems.com/products/ea/downloads.html>.

Other Read-Only Options

You can also make your model available to others in a read-only format by:

- Running Enterprise Architect over a VPN connection
- Generating an HTML report on the model, which can be published on the web with read-only access (see *Report Creation in UML Models*).

1.1.6 Sparx Systems MDG Add-Ins

Enterprise Architect is the core for a range of Model Driven Generation (MDG) Add-Ins that enable you to extend its modeling capabilities to use more specialized, niche frameworks and profiles. Some of these, such as Archimate, BPEL, BPMN, Data Flow Diagrams, Eriksson-Penker Extensions, ICONIX, Mind Mapping, Systems Modeling Language (SysML) and SPEM are already provided with the Enterprise Architect installer (see *Extending UML With Enterprise Architect*).

Enterprise Architect provides support for downloading MDG Technologies from external system files or websites, or for creating your own easily with the Enterprise Architect MDG Technology Wizard. For further information, see *SDK for Enterprise Architect*.

Sparx Systems also market a number of MDG products, as follows:

- MDG Technology For:
 - Zachman Framework
 - The Open Group Architecture Framework (TOGAF)
 - Department Of Defense Architecture Framework - Ministry Of Defence Architecture Framework (DoDAF-MODAF)
 - Data Distribution Service (DDS)
 - Python (for Enterprise Architect versions 4.5 to 5.0, integrated in later versions) (* free product! *)
 - CORBA (* free product! *)
 - Java Beans (* free product! *)
 - Testing (* free product! *)
- MDG Integration For:
 - Eclipse 3.3
 - Visual Studio 2005 and 2008
 - Siemens PLM Teamcenter Systems Engineering (TcSE)
- MDG Link For
 - Eclipse
 - Visual Studio.Net
 - Microsoft Visio (* free product! *)
 - Telelogic DOORS

Over time, this list is being extended to include further products.

Sparx Systems provide different editions of Enterprise Architect tailored for [systems engineering](#)^[13] and [business engineering](#)^[12], or [both](#)^[12] together. These editions incorporate several of the above MDG Technologies and other add-ins.

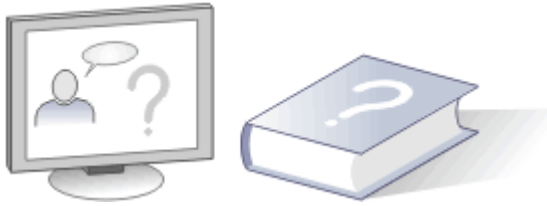
Product Information

For the latest list of available Add-Ins and an introduction to each product, including details of pricing, purchasing and download options, see the [Products Page on the Sparx Systems website](#). When you purchase one of the Add-Ins, you receive one or more license keys and instructions on obtaining, installing and registering the product.

The information page for most products provides a link to download the product **User Guide** in .pdf format.

The product User Guide can also be displayed as a .chm file online within the product itself. To access this online help in Enterprise Architect, select the **Add-Ins | <productname> | Help** menu option.

1.2 Help and Support



Enterprise Architect has three main help and information systems to assist you in using the product:

- **Tasks Pane**
- Enterprise Architect Help
- The Sparx Systems website.

In addition Sparx Systems recommend that you fully explore the sample project supplied with Enterprise Architect. It assists you in learning to use Enterprise Architect and offers tips on getting the most out of Enterprise Architect's features. Click on the **EAExample** option on the Enterprise Architect **Start Page** (see *Using Enterprise Architect - UML Modeling Tool*).


If you have purchased Enterprise Architect and are a registered user, you can also contact [Sparx Support](#) ^[17] to answer any queries or problems.

Tasks Pane

The Enterprise Architect **Tasks Pane** provides context-sensitive guidance, tools, demonstrations and other online resources to help you understand any area of Enterprise Architect that you are interested in. The **Tasks Pane** automatically displays on the right of the screen when you first open Enterprise Architect, showing the *Getting Started* topics. You can select other task areas by clicking on the **More tasks** option in the toolbar. For more information on the **Tasks Pane**, see *Using Enterprise Architect - UML Modeling Tool*.

Enterprise Architect Help

Enterprise Architect Help provides comprehensive documentation of Enterprise Architect and covers every aspect and facility of the product. To access Help within Enterprise Architect:

- Click on the Help icon () in the various toolbars
- Select the **Help | Help Contents** menu option
- Click on the **Help** button on a dialog (for Help specific to that dialog).

Enterprise Architect Help is extensive; if you cannot quickly locate the topic you require in the online contents list, you can use one of two search facilities:

- Click on the **Index** tab, type in a keyword or key phrase appropriate to the subject you require help for, and press **[Enter]**; double-click on the appropriate index item
- Click on the **Search** tab, type in a word or phrase to search for, and click on the **List Topics** button; double-click on the required topic.

The Enterprise Architect Help is also available separately from the product, in different formats. See the [Available Helpfile Formats](#) ^[17] topic.

Sparx Systems Website

The Sparx Systems website is also extensive, and provides information and announcements concerning the company and its full range of products, as well as tutorials, white papers, templates and solutions. It also provides a user forum, community site (see below) and support network; Sparx Systems are highly responsive to user feedback and requirements, and the web site enables rapid communication concerning problems, solutions and enhancements.

You can access the web page and user forum within Enterprise Architect from the **View | Other Project Tools | Internal Web Browser** menu option, and through the **Online Resources** topics in the **Tasks Pane**.

If you do not have Enterprise Architect open, the Sparx Systems website address is <http://www.sparxsystems.com/>.

The user forum address is www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi.

Community Site

The Sparx Systems website also hosts the [Sparx Systems Enterprise Architect Community Site](#). This is a central location for the Enterprise Architect community to publish resources and share experiences. From this site you can download the latest news, tutorials, resources, best practices, tips, techniques and user-generated content for Enterprise Architect. You can also, as a registered author, contribute content and share your expertise with the wider community.

1.2.1 Available Helpfile Formats

You can access the latest Enterprise Architect help files from the following locations:

- **.CHM** format: www.sparxsystems.com/bin/EA.chm
- **.CHM** format inside a **.ZIP** file: www.sparxsystems.com/bin/EAHelp.zip
- **.PDF** format: www.sparxsystems.com/bin/EAUserGuide.pdf
- **.HTML** format: www.sparxsystems.com/EAUserGuide/index.html

Version and release date information for the help files can be found at:

- www.sparxsystems.com/ea_downloads.htm#Helpfiles, or
- www.sparxsystems.com/registered/reg_ea_down.htm#Helpfiles (registered users).

1.2.2 Support

Technical support for Enterprise Architect is available to registered users. Responses to support queries are sent by email. Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

Registered users can lodge a support request, by visiting:
http://www.sparxsystems.com/registered/reg_support.html.

Trial users can contact Sparx Systems with questions regarding their evaluation at:
support@sparxsystems.com.

An online user forum is also available for your questions and perusal, at
<http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>.

1.3 Formal Statements



Please take the time to read the following legal statements concerning Sparx Systems Enterprise Architect:

- [Software Copyright Notice](#)^[18]
- [Enterprise Architect End User Licensing Agreement](#)^[18]
- [Acknowledgement of Trademarks](#)^[21]

Sparx Systems would also like to gratefully [acknowledge contributions](#)^[22] to the development of Enterprise Architect.

1.3.1 Copyright Notice

Copyright © 1998 - 2010 Sparx Systems Pty. Ltd. All rights reserved.

The software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the [license agreement](#)^[18] for full details.

Due to continued product development, this information can change without notice. The information and intellectual property contained herein is confidential between Sparx Systems and the client and remains the exclusive property of Sparx Systems. If you find any problems in the documentation, please report them to us in writing. Sparx Systems does not warrant that this document is error-free. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Sparx Systems. Licensed users are granted the right to print a single hardcopy of the user manual per licensed copy of the software, but may not sell, distribute or otherwise dispose of the hardcopy without written consent of Sparx Systems.

Sparx Systems Pty. Ltd.

7 Curtis St,
Creswick, Victoria 3363,
AUSTRALIA

Phone: +61 (3) 5345 1140
Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com
Sales Email: sales@sparxsystems.com

Website: www.sparxsystems.com

Scintilla and SciTE

Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org> All Rights Reserved.

Permission to use and distribute this (Scintilla) software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

1.3.2 End User License Agreement

Enterprise Architect Modeling Tool Version 8.0

Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering & Ultimate Editions

Copyright (C) 1998-2010 Sparx Systems Pty Ltd. All Rights Reserved

IMPORTANT- READ CAREFULLY: This End User License Agreement ("EULA") is a legal agreement between YOU as Licensee and SPARX SYSTEMS ("SPARX") for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA. If YOU do not agree to the terms of this EULA, promptly return the unused SOFTWARE PRODUCT to the place of purchase for a full refund.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd A.C.N 085 034 546. Subject to the terms of this EULA, YOU are granted a non-exclusive right for the duration of the EULA to use the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

Your use of this software indicates your acceptance of this EULA and warranty.

DEFINITIONS

In this End User License Agreement, unless the contrary intention appears:

- "ACADEMIC EDITION" means an edition of the SOFTWARE PRODUCT purchased for educational purposes at an academic discount price.
- "EULA" means this End User License Agreement.
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546.
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA.
- "Registered Edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: http://www.sparxsystems.com/ea_purchase.htm.
- "SOFTWARE PRODUCT" or "SOFTWARE" means Enterprise Architect, UML Case Tool, Desk Top, Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, which includes computer software and associated media and printed materials, and may include online or electronic documentation.
- "Support Services" means email based support provided by SPARX, including advice on usage of Enterprise Architect, investigation of bugs, fixes, repairs of models if and when appropriate, and general product support.
- "SPARX support engineers" means employees of SPARX who provide on-line support services.
- "Trial edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available free of charge for evaluation purposes for a period of thirty (30) days.
- "EA LITE" means the LITE version of Enterprise Architect that is distributed free of charge as a read-only viewer of .EAP files.

GRANT OF LICENSE

In accordance with the terms of this EULA YOU are granted the following rights:

- a) To install and use one copy of the SOFTWARE PRODUCT or, in its place, any prior version for the same operating system, on a single computer. As the primary user of the computer on which the SOFTWARE PRODUCT is installed, YOU may make a second copy for your exclusive use on either a home or portable computer.
- b) To store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT over an internal network. If YOU want to increase the number of users entitled to concurrently access the SOFTWARE PRODUCT, YOU must notify SPARX and agree to pay an additional fee.
- c) To make copies of the SOFTWARE PRODUCT for backup and archival purposes only.

EVALUATION LICENSE

The Trial version of Enterprise Architect is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of thirty (30) days.

Upon expiration of the thirty (30) days, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of Enterprise Architect after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

SPARX may extend the evaluation period on request and at their discretion.

If YOU choose to use this software after the 30 day evaluation period a license must be purchased (as described at http://www.sparxsystems.com/ea_purchase.htm). Upon payment of the license fee, YOU will be sent details on where to download the registered edition of Enterprise Architect and will be provided with a suitable software 'key' by email.

EA LITE

Subject to the terms of this Agreement [EA LITE](#) may be installed on any machine indefinitely and free of charge. There are no fees or Sparx support services in relation to EA LITE.

ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell, rent, lease, translate, adapt, vary, modify, decompile, disassemble, reverse engineer, create derivative works of, modify, sub-license, loan or distribute the SOFTWARE PRODUCT other than as expressly authorized by this EULA.

YOU further undertake not to reproduce or distribute license key-codes except under the express and written permission of SPARX.

If the Software Product purchased is an Academic Edition, YOU ACKNOWLEDGE THAT the license is limited to use in an educational context, either for self-education or use in a registered teaching institution. The Academic Edition may not be used to produce commercial software products or be used in a commercial environment, without the express written permission of SPARX.

ASSIGNMENT

YOU may only assign all your rights and obligations under this EULA to another party if YOU supply to the transferee a copy of this EULA and all other documentation including proof of ownership. Your license is then terminated.

TERMINATION

Without prejudice to any other rights, SPARX may terminate this EULA if YOU fail to comply with the terms and conditions. Upon termination YOU or YOUR representative shall destroy all copies of the SOFTWARE PRODUCT and all of its component parts or otherwise return or dispose of such material in the manner directed by SPARX.

WARRANTIES AND LIABILITY

WARRANTIES

SPARX warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and any Support Services provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX support engineers will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

EXCLUSIONS

To the maximum extent permitted by law, SPARX excludes, for itself and for any supplier of software incorporated in the SOFTWARE PRODUCT, all liability for all claims, expenses, losses, damages and costs made against or incurred or suffered by YOU directly or indirectly (including without limitation lost costs, profits and data) arising out of:

- YOUR use or misuse of the SOFTWARE PRODUCT
- YOUR inability to use or obtain access to the SOFTWARE PRODUCT
- Negligence of SPARX or its employees, contractors or agents, or of any supplier of software incorporated in the SOFTWARE PRODUCT, in connection with the performance of SPARX' obligations under this EULA, or
- Termination of this EULA by either party for any reason.

LIMITATION

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality, fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in this EULA. However, the liability of SPARX for any breach of such term shall if permitted by legislation be limited, at SPARX's option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

- If the breach relates to the SOFTWARE PRODUCT:
 - the replacement of the SOFTWARE PRODUCT or the supply of an equivalent SOFTWARE PRODUCT

- the repair of such SOFTWARE PRODUCT
- the payment of the cost of replacing the SOFTWARE PRODUCT or of acquiring an equivalent SOFTWARE PRODUCT, or
- the payment of the cost of having the SOFTWARE PRODUCT repaired.
- If the breach relates to services in relation to the SOFTWARE PRODUCT:
 - the supplying of the services again, or
 - the payment of the cost of having the services supplied again.

TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and licenses.

Windows®, Windows 98, Windows NT, Windows ME, Windows XP, Windows Vista, Windows 2000 and Windows 2003 Server are trademarks of Microsoft®.

GOVERNING LAW

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA, in the state of Victoria.

1.3.3 Trademarks

Trademarks of Microsoft

- Microsoft Word
- Microsoft Office
- Windows®
- ActiveX

Registered Trademarks of The OMG

- CORBA®
- the OMG Object Management Group logo
- The Information Brokerage®
- CORBA Academy®
- IIOP®
- XMI®
- UML®
- The UML Cube logo

Trademarks of The OMG

- OMG™
- Object Management Group™
- The CORBA logo
- ORB™
- Object Request Broker™
- The CORBA Academy design
- OMG Interface Definition Language™
- IDL™
- CORBAservices™
- CORBAfacilities™
- CORBAmed™
- CORBAnet™
- Unified Modeling Language™
- MOF™
- CWM™

- Model Driven Architecture™
- MDA™
- OMG Model Driven Architecture™
- OMG MDA™

1.3.4 Acknowledgements

Some parts of this application include code originally written by various authors and modified for use in Enterprise Architect.

Marquet Mike

Print listview contents
mike.marquet@altavista.net

Davide Pizzolato

CXImage Library
© 7-Aug-2001
ing.davide.pizzolato@libero.it

Neil Hodgson

Scintilla editor
© 1998-2003
neilh@scintilla.org

Also, many thanks to all those who have made suggestions, reported bugs, offered feedback and helped with the beta-testing of Enterprise Architect. Your help has been invaluable.

1.4 If You Have the Trial Version



If you are evaluating the Enterprise Architect trial version, note that the software operates for a limited period. To continue using Enterprise Architect when the trial period expires, you can either apply to extend the trial period (see [below](#)^[23]), or purchase and register a full license as explained in the following topics:

- [Order Enterprise Architect](#)^[25]
- [Installation](#)^[25]
- [Register a Full License](#)^[26]

If you already have a full license edition of Enterprise Architect and want to register Add-Ins or upgrade to the Professional, Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions, see the [License Management](#)^[54] topic.

Try Out Editions

When you start up the Enterprise Architect trial version, the **Select Trial Version** prompt displays:

A screenshot of a dialog box titled "Select Trial Version". The dialog box has a white background and a blue border. At the top, it says "To help you determine the edition best suited to your needs, you can open this trial product in any configuration." Below that, it says "Please select which trial edition you wish to start". There are two columns of options. The left column is titled "Select Trial Edition" and has six radio buttons: Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering, and Ultimate. The "Ultimate" radio button is selected. The right column is titled "Videos" and has three blue hyperlinks: "EA Overview", "EA Walkthrough", and "More Video Tutorials". Below the "Videos" section is a "Links" section with three blue hyperlinks: "Compare Editions", "Introduction to EA", and "User Forum". At the bottom left, there is a "Select Workspace Layout" section with a dropdown menu showing "Basic Layout". At the bottom right, there are three buttons: "OK", "Cancel", and "Help".

This enables you to select the edition of Enterprise Architect to evaluate, by clicking on the appropriate radio button. The prompt also directs you to useful information such as a comparison of the different editions, or a walkthrough of the Enterprise Architect facilities. When using Enterprise Architect, you can select from a range of workspace layouts; the **Select Trial Version** prompt enables you to select the initial layout to use.

Extend Trial Period

If you are testing the trial version and need more than 30 days to evaluate it, you can apply to Sparx Systems Sales (sales@sparxsystems.com) for an extension of the trial period. This extension can be an additional 30 to 150 days; if you do not request a long enough extension in the first place, you can submit further requests. Sparx Systems Sales send you an extension key by email.

Note:

The trial period must expire before you can enter the extension key.

To extend the trial period, after receipt of your extension key, follow the steps below:

1. Open the Enterprise Architect trial version. The **Evaluation Version of Enterprise Architect** dialog displays.

**Note:**

Once the trial period has expired, you cannot proceed beyond this dialog without extending the trial period.

2. Press **[Ctrl]** whilst you click on the **Continue Trial** button. The **Upgrade Key** dialog displays.

3. In the **Upgrade Key** field, type or copy-and-paste the extension key you received from Sparx Systems Sales.
4. Click on the **OK** button. Enterprise Architect confirms that your trial period has been extended. Your trial period is extended by the period of days stated in the email from Sparx Systems Sales. You can now *restart* and use the Trial version of Enterprise Architect.

1.4.1 Order Enterprise Architect

Enterprise Architect is designed, built and published by Sparx Systems and available from [Sparx Systems](#).

The trial version of Enterprise Architect is identical to the registered edition with the exception that all diagrams are output to files with an embedded watermark. The trial software stops working after the trial period has elapsed. On purchase of a suitable license or licenses, the registered version is made available for download.

The latest information on pricing and purchasing is available at: [Sparx Systems Purchase/Pricing Website](#).

Purchase Options

- On-line using a secure credit-card transaction; see: [Pricing and Purchase Options](#)
- Fax
- Check or equivalent
- Bank transfer.

For more information, contact sales@sparxsystems.com.

1.4.2 Installation

Enterprise Architect is distributed as a single executable setup file (.exe). After you execute this file, Enterprise Architect is immediately available to create projects as .EAP files.

If you plan to use the Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition with SQL Server, MySQL, PostgreSQL, Access 2007, Sybase Adaptive Server Anywhere or Oracle 9i, 10g or 11g as a model repository (see below), you require additional files and supplementary installation processes. Please note that installation and maintenance of these database management systems is not covered under the support agreement.

The latest evaluation and registered versions of Enterprise Architect are always available from the [Sparx Systems](#) website. The registered version is available through the registered user area of the web site, which requires a username and password to access. These are provided upon purchase of a license.

System Requirements

The system requirements for installing Enterprise Architect are defined on the [Enterprise Architect | System Requirements](#) page of the Sparx Systems website.

Windows Vista

Under Windows Vista (with User Account Control turned on) an application starts with only Standard permissions, regardless of what level of authority the current user has. As a result, an installer run normally with an Admin account under Vista only has Standard privileges and either is not able to write to certain critical areas of the registry/file system, or redirects the write requests to a per-user virtualized registry/file system.

Sparx Systems recommend that if you are installing on Windows Vista, always run the Enterprise Architect installer with Administrator privileges (right-click on the downloaded installer icon and select the **Run as administrator** menu option).

Install Enterprise Architect

Run the Enterprise Architect setup program. Generally you can accept all the default options without change.

To place Enterprise Architect in a directory other than the default, enter the name of the destination when prompted.

You might be prompted to restart your computer when the installation completes. Although this is not always necessary (if you already have the components Enterprise Architect requires installed on your computer), you should restart just to be certain.

If you intend to run Enterprise Architect on Linux, refer to the [Installation and Use](#) page on the Sparx Systems website. To support the Scripting facility in the Corporate and extended editions of Enterprise Architect, you must also install Internet Explorer 6.0 or later revisions.

Corporate, Business and Software Engineering, Systems Engineering and Ultimate edition users planning to use SQL Server, MySQL, PostgreSQL, Sybase Adaptive Server Anywhere, Access 2007 or Oracle 9i, 10g or 11g as their model repository can access scripts that create the required data structures for the choice of DBMS. You can find these at one of the following pages:

- The Corporate edition [Resources](#) page
- The Trial Corporate edition [Resources](#) page.

Note:

Enterprise Architect requires *Read/Write* access to the program files directory where Enterprise Architect has been installed.

1.4.3 Register a Full License

The trial version of Enterprise Architect available for download is an evaluation version only. For the full version you must first purchase one or more licenses. The license code supplied determines which edition (Desktop, Professional, Corporate, Business and Software Engineering, System Engineering or Ultimate) is activated after installation.

The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions can be activated by either a private license key or a shared (or floating) license key. Shared license keys, for a user population that includes temporary users, are checked out of and checked in to a Sparx Systems keystore.

Private license keys and shared license keys have different formats, so you cannot use one in place of the other.

Register Enterprise Architect

To obtain the full version and complete the registration process, follow the steps below:

1. Purchase one or more licenses.
Once you have paid for a licensed version of Enterprise Architect, you receive (via email or other suitable means):
 - a license key or keys
 - the address of a web site from which to download the full version
 - if you have purchased a Sparx Systems key store, the address of a web site from which to download the keystore.
2. Save the license key(s) and download the latest full install package from the address supplied.
3. Run the setup program to install the full version.
4. Open Enterprise Architect from the **Start Menu** or desktop icon. The **License Management** dialog displays.

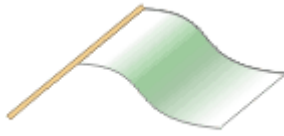


5. Click on the **Add Key** button. The **Add Registration Key** dialog displays.
6. Refer to the [Add License Key](#) topic.
7. Click on the **OK** button. The full version is now activated on your PC, and Enterprise Architect displays the message: *Registration succeeded! Thank you for purchasing Enterprise Architect <type> Edition.*

See Also

- [Upgrade an Existing License](#)^[59]
- [Register Add-In](#)^[62]

2 Start Modeling



This guide provides two options to help you gain an understanding of how to perform UML modeling with Enterprise Architect:

- A brief outline of the types of [work and tasks](#)^[8] that Enterprise Architect supports, so that you can quickly locate the more detailed explanations of subjects that interest you
- A Quick Start tutorial to Enterprise Architect that illustrates how to open and create new projects, navigate Enterprise Architect, and use Enterprise Architect to perform various tasks in system and process modeling. It leads on to examining the Enterprise Architect User Interface (in *Using Enterprise Architect - UML Modeling Tool*) and work areas applicable to certain [Project Roles](#)^[38].

At various points throughout the Enterprise Architect Help, there are further Quick Start topics and sections to help you use the system immediately to experiment with a feature of Enterprise Architect. Use the Help [Index](#) tab and search for *Quick Start* to locate these topics.

2.1 Quick Start - Create a Project



Tutorial

Welcome to Enterprise Architect! This quick-start tutorial helps you start UML modeling with Enterprise Architect.

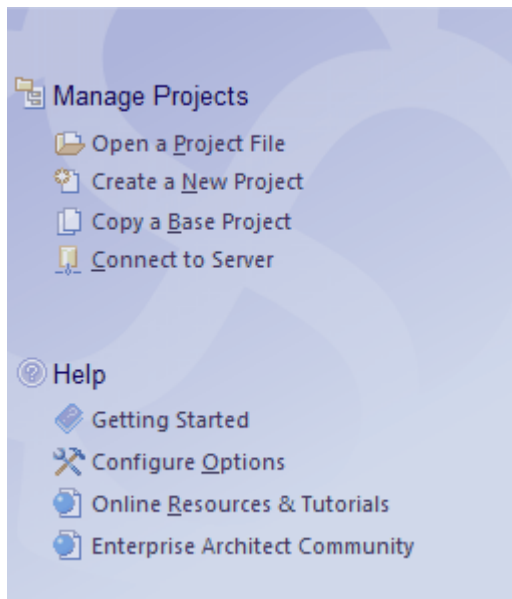
As you read through the Quick Start sections, have Enterprise Architect open so that you can explore and try out the functions described. By the end of the Quick Start tutorial you should be able to begin modeling your own software projects with Enterprise Architect and UML.

The tutorial guides you through creating a simple project. Throughout the descriptions there are hyperlinks to more detailed information on a range of topics. Follow these links if you would like more information, or ignore them if you want to just follow the steps.

Your task is to create a new project (see *UML Model Management*) and then add a View, a package, a diagram, elements and connectors.

Create a Project

When you start Enterprise Architect it opens at the Start Page (see *Using Enterprise Architect - UML Modeling Tool*):



1. Click on the **Create a New Project** option. The **New Project** dialog displays.
2. In the **File name** field, type a meaningful name for the project and click on the **Save** button to create the project file. The **Select Model(s)** dialog displays.
3. You now select one or more model templates (these provide you with the basic structures - packages and diagrams - for your project, as well as references to useful help files to get you started). Select the checkbox of each model that interests you.
4. Click on the **OK** button. Enterprise Architect creates a Model Package for each selected template and displays it in the **Project Browser**, on the right-hand side of the screen.

Note:

- For more information on the model templates and **Project Browser**, see *Using Enterprise Architect - UML Modeling Tool*.
- You could also quickly create a project by copying an existing base project provided with Enterprise Architect; see the *Copy a Base Project* topic in *UML Model Management*.

Expand The Project

To navigate through your project, in the **Project Browser** click on the 'plus' icon against each folder or *package* to expand it.

Double-click on the *diagram* icon displayed underneath a package name. Enterprise Architect displays the sample diagram for that model in the **Diagram View**, which is in the middle of the screen.

Add a View To Your Model

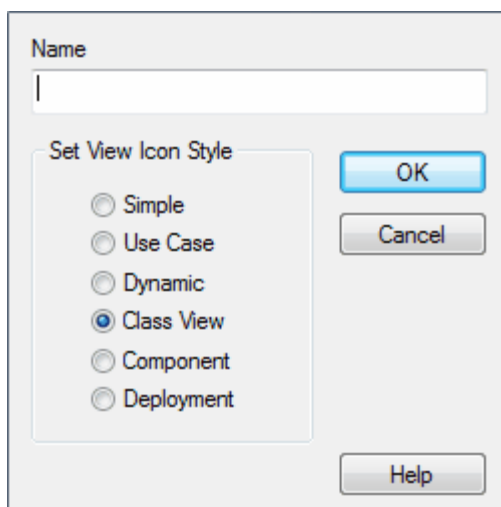
Now that you have created a project containing at least one model, you can add a **View**^[30] to a model, and then add a package with diagram, elements and connectors (relationships).

2.1.1 Add a View to a Model

A View is the highest-level container, or package, within a model. There are six types of View, five of which represent conventional ways of categorizing the structures or purposes of a model, and one (*Simple View*) for developing your own categorization.

To create a View, follow the steps below:

1. Right-click on your model name in the **Project Browser**. The context menu displays.
2. Select the **New View** menu option. The **Create New View** dialog displays.



3. In the **Name** field, type the name of the View.
4. In the **Set View Icon Style** panel, click on the radio button for the type of View to create.
5. If the model root node had been under version control, an **Add to Version Control** checkbox would display, defaulted to selected. Ignore this for now.
6. Click on the **OK** button.

For information on Version Control, see *Version Control Within UML Models using Enterprise Architect*.

Add a Package To Your Model

Now that you have created a View in the model, you can add a **package**^[31] and diagram to that View or any other in the model, and then add elements and connectors (relationships).

2.1.2 Add a Package To a Model

A *Package* is a container of model elements, and is displayed in the **Project Browser** as the 'folder' icon familiar to Windows users. Package contents are arranged alphabetically.

In the **Project Browser** click on a package and, in the **Project Browser** toolbar, click on the **New Package** icon



Enterprise Architect displays a prompt for the package name.

Note:

This prompt also contains the **Automatically add new diagram** option for automatically creating a diagram for the package, which defaults to selected. This is very a useful feature, but for the purposes of this introduction deselect the checkbox against the option.

Type in a name and click on the **OK** button. Enterprise Architect adds the new package subordinate to the package you selected.

Add a Diagram To a Package

Now [add a diagram](#) .

Additional Information

For additional information on packages, and adding packages and Views (top-level packages), see the *Work With Packages* and *Add a Package* topics in *UML Modeling with Enterprise Architect – UML Modeling Tool* and *Add Additional Views* topic in *UML Model Management*.

2.1.3 Add a Diagram to a Package

A diagram is a representation of the components or elements of your model and, depending on the type of diagram, how those elements are connected or how they interact.

When you first create a project, Enterprise Architect provides simple examples of diagrams appropriate to your selected model patterns, with annotations. You can edit these diagrams, but here we create an additional one.

Click on your new package and, in the **Project Browser** toolbar, click on the **New Diagram** icon .

The **New Diagram** dialog displays.

Note:

When you create a package, if you leave the **Automatically add new diagram** option selected, the **New Diagram** dialog displays automatically.

Click on a diagram category in the **Select From** panel, and a diagram type in the **Diagram Types** panel, then click on the **OK** button. Enterprise Architect adds a diagram object to the package, with the same name as the package. It also opens the **Diagram View** for your diagram, in the center of the screen.

Add Elements to a Diagram and a Package

Now [add some elements](#) .

Additional Information

For additional information on diagrams and adding them to a project, see *UML Diagrams* in *The UML Dictionary* and *Add New Diagrams* in *UML Modeling with Enterprise Architect – UML Modeling Tool*.

2.1.4 Add Elements

You have several options for adding elements (the UML model building units) to a package and/or diagram.

The simplest method is to use the Enterprise Architect UML **Toolbox** to the left of the diagram, which automatically lists the elements applicable to the type of diagram you have created. Just click on the required element and drag it onto your diagram.

Two things might occur before the element displays on the diagram:

- If you have selected an *Object* element, Enterprise Architect prompts you to define what stereotype the object is based on (an object can represent a wide range of things, and a stereotype helps you define what the object or element is); for now, select any value.
- The element [Properties](#) ^[33] dialog displays. If it does not display, double-click on the element on the diagram.

You can use the **Properties** dialog to define the characteristics of the element, such as its name. Type a name in the **Name** field, and click on the **OK** button. Look at the **Project Browser**, underneath the package in which you created the diagram. The element is listed.

Tip:

Enterprise Architect has two very useful features:

- To find out more about the type of element you have dragged on to a diagram, right-click on the element and select the **UML Help** menu option. This displays a Help page on the element type.
- If you are creating several elements of one type, after creating the first just press **[Shift]+[F3]** or **[Ctrl]+click** to create the next element of that type.

You can also drag or paste existing elements onto a diagram from the **Project Browser**. This enables you to make use of previous work in defining elements.

Add Connectors Between Elements

Now [connect the elements](#) ^[32] with relationships.

Additional Information

When you click on an element on a diagram, a number of icons display outside the top right corner. These are the Quick Linker and Toolbar icons that enable you to quickly add more elements and connectors and to operate on the elements. You can read about and experiment with these later.

For additional information on elements and adding elements to a project, see *UML Elements* in *The UML Dictionary*, and *Create Elements* and *Paste From the Project Browser* in *UML Modeling With Enterprise Architect - UML Modeling Tool*.

2.1.5 Add Connectors

Connectors define specific relationships between specific elements, so you usually create them directly on the diagram by dragging the required relationship type from the Enterprise Architect UML **Toolbox**. As for elements, the **Toolbox** automatically presents the connector or relationship types appropriate to the type of diagram.

Create two elements on the diagram. Click on a connector in the **Toolbox**, click on the source element in the relationship, then drag across to the target element. This creates the selected connection between the two elements. If you double-click on the connector, the connector [Properties](#) ^[33] dialog displays, and you can define the characteristics of the relationship.

Tip:

Enterprise Architect has three very useful features:

- To find out more about the type of connector you have dragged on to a diagram, right-click on the connector and select the **UML Help** menu option. This displays a Help page on the connector type.
- If you are creating several connectors of one type, after creating the first just click on the appropriate source element and press **[F3]** to create the next connector of that type.
- As you drag a connector, you can press **[Shift]** to create a bend in the connector. If necessary, you can put several bends in the connector line, pressing **[Shift]** every time you want to change direction. To roll back the bends, keep holding the left mouse button down and press **[Backspace]** as many times as is necessary.

Moving and Deleting Elements and Connectors

Having created a model with some components, you can [move](#)^[35] those components around and [delete](#)^[36] them. You should also know how to [save](#)^[37] your work.

Additional Information

For additional information on connectors, see *The UML Dictionary*. For further information on creating a connector through the **Project Browser**, or with the Quick Linker, see the *Create Connector* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool* and the *Create Connections Between Elements* topic in *Using Enterprise Architect - UML Modeling Tool*.

2.1.6 Define Properties

When you create an element and connect it to another element, you usually have to define various characteristics of both the element and the connector to identify the purpose and function they represent. You do this using a **Properties** dialog.

Enterprise Architect is initially configured to display the **Properties** dialog automatically when you create an element or connector, but it is easy (and often convenient) to turn the dialog display off (see the *Properties* topic in *Using Enterprise Architect - UML Modeling Tool*). If the default display has been turned off, you can display the dialog by:

- double-clicking on the element or connector in the diagram or
- right-clicking on it in the **Project Browser** and selecting the **Properties** menu option.

Properties dialogs vary between element types and between elements and connectors but, as you saw when you created your first element, they look something like this:

When you create elements, Enterprise Architect automatically names and numbers them by type - for example, Class1, Class2 - so you should at least change the **Name** field to more easily identify each element.

Enterprise Architect does not automatically name connectors, but for many connector types you should provide a name that describes the purpose of the connection.

See the *Element Properties* topic and the *Connector Properties* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool* for a full description of the element **Properties** dialog and the connector **Properties** dialog.

Explore User Interface

So far you have been using the **Project Browser** and **Diagram View** to develop your project. At this point you should find out a bit more about the other facilities of the Enterprise Architect User Interface in *Using Enterprise Architect - UML Modeling Tool*.

When you have finished exploring the User Interface topics, go to [Quick Start - Project Tasks](#)^[38] to identify areas of Enterprise Architect that provide particular support for your job role.

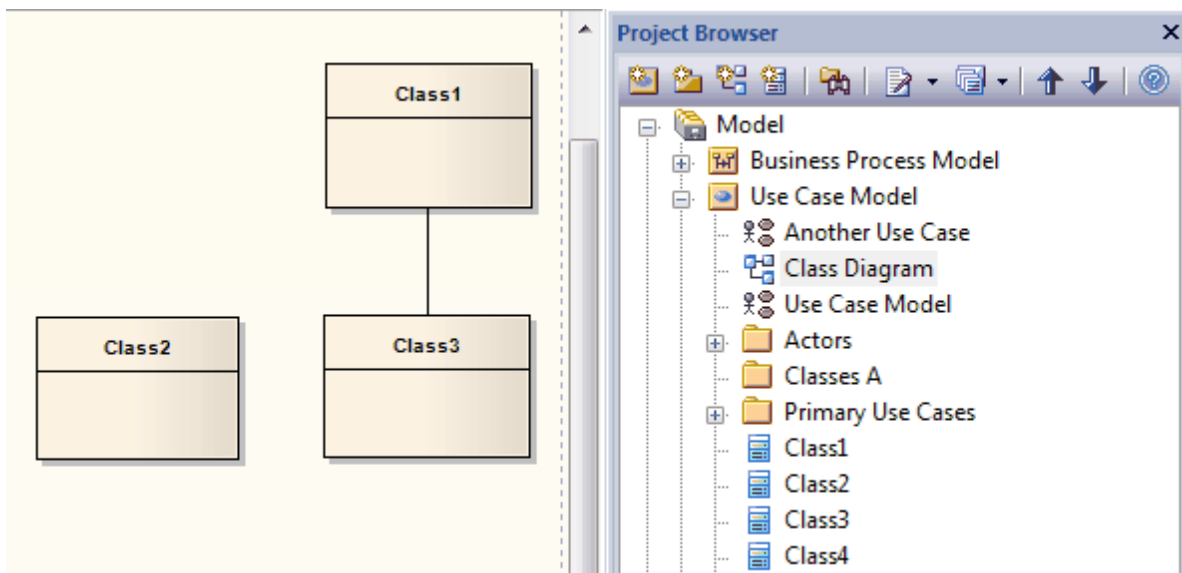
2.1.7 Move Components

You have created a project containing packages, diagrams and elements, and you have connected the elements. You might have arranged your components in the wrong project structure. How do you change where things are?

Note:

Changing [names and properties](#) ^[33] is discussed a little later.



In this topic, the explanations refer to the following example:



Notes:

- You display and work on your model in the **Project Browser**, and display and work on a diagram in the **Diagram View**.
- In the **Project Browser**, the contents of a package are listed in the order *diagrams | child packages | elements*. Elements are further arranged in type order. Within their types, components are initially listed in alphabetical or numerical order.
- Moving an element or package has no effect on any relationships that the element, package, or elements within the package have. You have to specifically create, delete or move the relationships themselves.

Move Components Within a Package in the Project Browser

To move a diagram, child package or element within its parent package, click on it in the **Project Browser** and click on  or  in the toolbar at the top of the window.

You could move *Class3* in the **Project Browser** above *Class1*, or move the *Actors* package underneath *Classes A*.

To revert to listing components in alphabetical order, right-click on the package and select the **Contents | Reset Sort Order** menu option.

Move Components Between Packages in the Project Browser

You might have created a diagram, child package or element in the wrong place in the **Project Browser**. To move a model component to another package, click on the component and drag it to the new package. This can be at either a higher level or a lower level.

You might, for example, drag *Class1* from the *Use Case Model* package into the *Business Process Model* package. *Class1* then is listed in the *Business Process Model* package in the **Project Browser**. As a similar example, you could drag *Class Diagram* into the *Business Process Model* package.

Moving elements in the **Project Browser** does not affect the use of elements in diagrams. In our example, *Class1* is initially in a diagram in the *Use Case Model* package. When you move *Class1* in the **Project Browser** from *Use Case Model* to *Business Process Model*, it still shows in the diagram in *Use Case Model*, and does not display in any diagram in *Business Process Model*.

Note:

Moving a diagram generally does not affect the location of elements in packages. If you move the *Class Diagram* out of *Use Case Model* into *Business Process Model*, all the elements in the diagram remain in the *Use Case Model* package.

However, elements of certain types might be used only within one diagram, have no meaning outside that diagram, and never be re-used in any other diagram. Such elements include Decision, Initial and Final Node elements. Therefore, if you move a diagram containing these elements, they **are** moved to the new parent package with the diagram.

To remove *Class1* from the *Use Case Model* diagram, click on it on the diagram and [delete](#) ^[36] it. Nothing happens to the element in the **Project Browser**. To put *Class1* into a diagram in the *Business Process Model* package, open the diagram in that package and drag the element from the *Business Process Model* package in the **Project Browser** onto the diagram.

Move Elements in a Diagram

If an element is not in the right position in the diagram, just click on the middle of it and drag it to the correct place. In the diagram above, you might move *Class2* below *Class 3*, and move *Class3* to the left. The element brings its connectors with it.

To make fine adjustments, press **[Shift]+[→]**, **[←]**, **[↑]** or **[↓]**.

Move Elements Between Diagrams

If an element is in the wrong diagram, you can delete it from that diagram and then drag it from the **Project Browser** into the right diagram.

Alternatively, select the element and *cut* it from the source diagram (**[Ctrl]+[X]**, or the **Cut** icon in the **Default Tools** toolbar) and then paste the element (**[Shift]+[Insert]** or the **Paste** icon in the **Default Tools** toolbar) into the target diagram.

You can also copy and paste elements between diagrams. In the source diagram, select the element and press **[Ctrl]+[C]** (or click on the **Copy** icon in the **Default Tools** toolbar), then in the target diagram press **[Shift]+[Insert]** (or click on the **Paste** icon in the **Default Tools** toolbar).

Move Connectors in a Diagram

You might have connected the wrong pair of elements. To move the end of a connector to a different element (for example, *Class2* instead of *Class3*), click on the end to display a black 'handle' box and drag the end to its new position. Be aware that the connector does not break from the original target element until the cursor is on the new target.

You can also tidy up a connection by dragging the end of the connector to a better position on the edge of the element, or move both ends at once by dragging the middle of the connector.

Additional Information

See the topics on [Deleting Components](#) ^[36] and [Saving Changes](#) ^[37].

For additional information on moving connectors and elements, see the *Arrange Connectors* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool* and the *Order Package Contents* topic in *Using Enterprise Architect - UML Modeling Tool*.

2.1.8 Delete Components

You can delete the components of a model from a diagram or from the **Project Browser**.

Delete From a Diagram

A diagram can contain elements, connectors, packages and other diagrams. To delete any of these from the diagram, click on it and press **[Delete]** on the keyboard.

Notes:

- Remember that the contents of the model are listed in the **Project Browser**. If you delete something from a diagram, it is not deleted from the **Project Browser**. This is because you can use the same component in several diagrams at once, so you only remove the representation of the component from a diagram.
- To delete a connector from a diagram, click on it and press **[Delete]**. This time, Enterprise Architect prompts you to select whether to delete the connector or just hide it. Unlike elements, the same connector is not reapplied in several places, so if you delete one it is removed completely from the model.
- Connectors can get confusing on a complex diagram, so it is useful to hide some of them to clarify a specific aspect of a more complex picture. To identify and reveal hidden connectors, see the *Links* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool*. However, first you should become more familiar with element and connector properties, through the [Quick Start - Define Elements and Relationship Properties](#) topic.

Delete From Project Browser

To delete a package, diagram or element from the **Project Browser**, right-click on the component and select the **Delete <name>** menu option.

For a package, this completely removes the package and all its contents - diagrams, child packages and elements - from the model.

For an element, this completely removes the element and its properties, connectors, child elements and child diagrams from the model, and from every diagram that contains it.

For a diagram, this completely removes the diagram and connectors from the model, but **not** the diagram's component elements. They remain in the parent package.

Additional Information


See the [Save Changes](#) topic.

For additional information on deleting elements, connectors and model views in Enterprise Architect, see the *Delete Elements* and *Delete Connectors* topics in *UML Modeling with Enterprise Architect – UML Modeling Tool*, and the *Delete Views* topic in *UML Model Management*.

2.1.9 Save Changes

Throughout much of your work in Enterprise Architect, any changes you make are automatically saved when you close the *dialog* (data entry window) on which you made the changes. In some cases the dialog contains a **Save** or **Apply** button, which enables you to save your changes and then keep working on the dialog.

If there is no specific dialog, such as when you create a diagram, you can save your work by:

- Clicking on the **Save** icon in the **Diagram** toolbar ()
- Pressing the **[Ctrl]+[S]** keyboard keys, or
- Selecting the **Diagram | Save** menu option.

Often, Enterprise Architect does not let you close a screen without confirming that you want to save or discard your changes.

You can also save your diagram changes automatically, by selecting the **Auto Save Changes** checkbox on the **Diagram Behavior** page of the **Options** dialog.

2.2 Quick Start - Project Tasks

Throughout a design and development project there are many different tasks to be performed, which could be carried out either by one person or - more probably - by members of a team with different responsibilities. In either case, Enterprise Architect supports most - if not all - of the responsibilities you might have on your project.

Therefore, the next topics to explore depend on the work you normally do on a project.

The descriptions below cover a number of job roles that Enterprise Architect supports. For those that most resemble your role on a project, follow the job title hyperlink to display a description of how that role might use Enterprise Architect, then follow links within those topics to explore some of the Enterprise Architect features of importance to the role.

Another area of responsibility that Enterprise Architect supports is System Administration - see *UML Model Management*.

Most of these roles work with specific types of diagram, so you might want to learn more about diagram types in general and specific types of diagram in particular. See the *UML Diagrams* topic in the *UML Dictionary*.

Several types of project team member might want to generate documentation on their work and reports on how the project is developing and changing. Enterprise Architect enables you to generate project documentation in either RTF or HTML format - see *Report Creation in UML Models*.

Note:

The Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect have a user security feature that can be applied or turned off. If security is turned on, you require the appropriate access permissions to use many of the Enterprise Architect facilities listed above. For further information, see the *List of Available Permissions* topic in *User Security in UML Models*.

Business Analyst

A [Business Analyst](#)^[41] might be responsible for modeling:

- Requirements
- High-level business processes
- Business activities
- Work flows
- System behavior.

Software Architect

A [Software Architect](#)^[43] might be responsible for:

- Mapping functional requirements of the system
- Mapping objects in real time
- Mapping the deployment of objects
- Defining deliverable components.

Software Engineer

A [Software Engineer](#)^[45] might be responsible for:

- Mapping Use Cases into detailed Classes
- Defining the interaction between Classes
- Defining system deployment
- Defining software packages and the software architecture.

Developer

A [Developer](#)^[46] might be responsible for:

- Forward, reverse and round-trip engineering
- Visualizing the system states
- Visualizing package arrangements
- Mapping the flow of code.

Technology Developer

A [Technology Developer](#)^[52] might be responsible for creating or customizing:

- UML Profiles
- UML Patterns
- Code Templates
- Tagged Value Types
- MDG Technologies
- Add-Ins.

Database Administrator

A [Database Administrator](#)^[53] might be responsible for:

- Developing databases
- Modeling database structures
- Creating logical data models
- Generating schema
- Reverse engineering databases.

Tester

A [Tester](#)^[49] might be responsible for:

- Developing test cases
- Importing requirements, constraints and scenarios
- Creating Quality Test documentation
- Tracking element defects and changes.

Project Manager

A [Project Manager](#)^[48] might be responsible for:

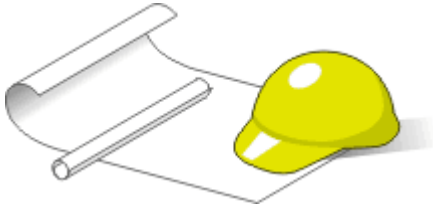
- Providing project estimates
- Resource Management
- Risk Management
- Maintenance Management.

Implementation Manager

An [Implementation Manager](#)^[50] might be responsible for:

- Modeling the tasks in rolling-out a project, including network and hardware deployment
- Assigning and tracking maintenance items on elements (issues, changes, defects and tasks).

3 Project Roles



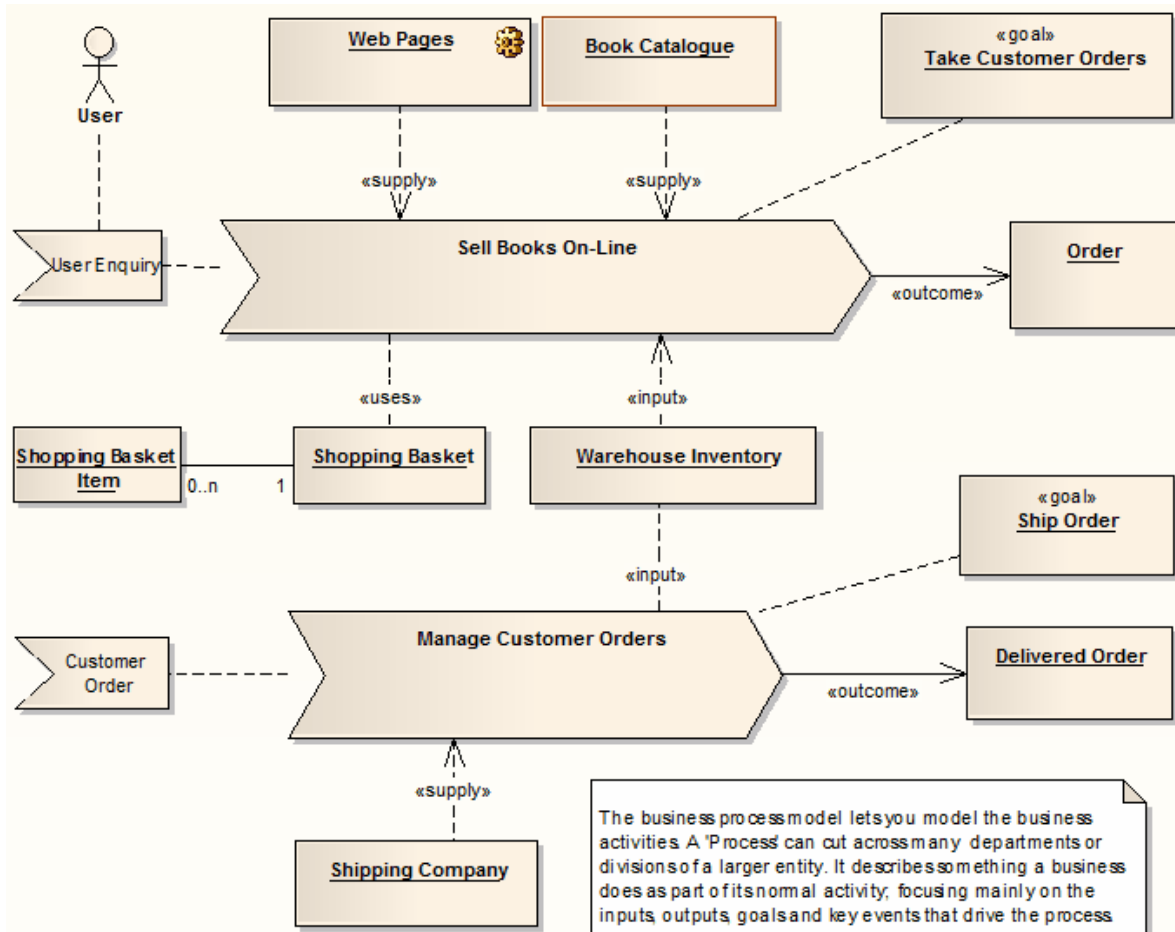
Enterprise Architect performs a number of tasks that are suited to a variety of professions. The way you use Enterprise Architect can depend on your role within a project. This topic describes some common working practices with Enterprise Architect for a range of project roles. There are tools for the roles of:

- [Business Analyst](#) ^[41]
- [Software Developer](#) ^[46]
- [Software Architect](#) ^[43]
- [Software Engineer](#) ^[45]
- [Project Manager](#) ^[48]
- [Tester](#) ^[49]
- [Database Administrator](#) ^[53]
- [Implementation Manager](#) ^[50]
- [Technology Developer](#) ^[52]

Click on the appropriate role title to explore how Enterprise Architect can assist you in carrying out your role within a model driven project.

3.1 Business Analysts

A Business Analyst can use Enterprise Architect to create high-level models of business processes (see the *Business Modeling* topic in *Extending UML With Enterprise Architect*. These include business requirements, activities, work flow, and the display of system behavior. Using Enterprise Architect, a Business Analyst can describe the procedures that govern what a particular business does. Such a model is intended to deliver a high-level overview of a proposed system.



Model High Level Business Processes

With Enterprise Architect the Business Analyst can model high level processes of the business with Analysis diagrams (see the *UML Dictionary*). Analysis diagrams are a subset of UML 2.1.1 Activity diagrams and are less formal than other diagram types, but they provide a useful means for expressing essential business characteristics and requirements.

Model Requirements

Gathering requirements is typically the first step in developing a solution, be it for developing a software application or for detailing a business process. It is an important step in the implementation of a project. Enterprise Architect enables you to define the Requirement elements, connect Requirements to the model elements for implementation, connect Requirements together into a hierarchy, report on Requirements, and move Requirements out of model element responsibilities. For more information, see *Requirements Management*.

Model Business Activities

The Business Analyst can use Activity diagrams (see the *UML Dictionary*) to model the behavior of a system and the way in which these behaviors are related to the overall flow of the system. Activity diagrams do not model the exact internal behavior of the system but show instead the general processes and pathways at a high level.

Model Work Flow

To visualize the cooperation between elements involved in the work flow, the Business Analyst can use an Interaction Overview diagram (see the *UML Dictionary*), which provides an overview of sub activities that are involved in a system.

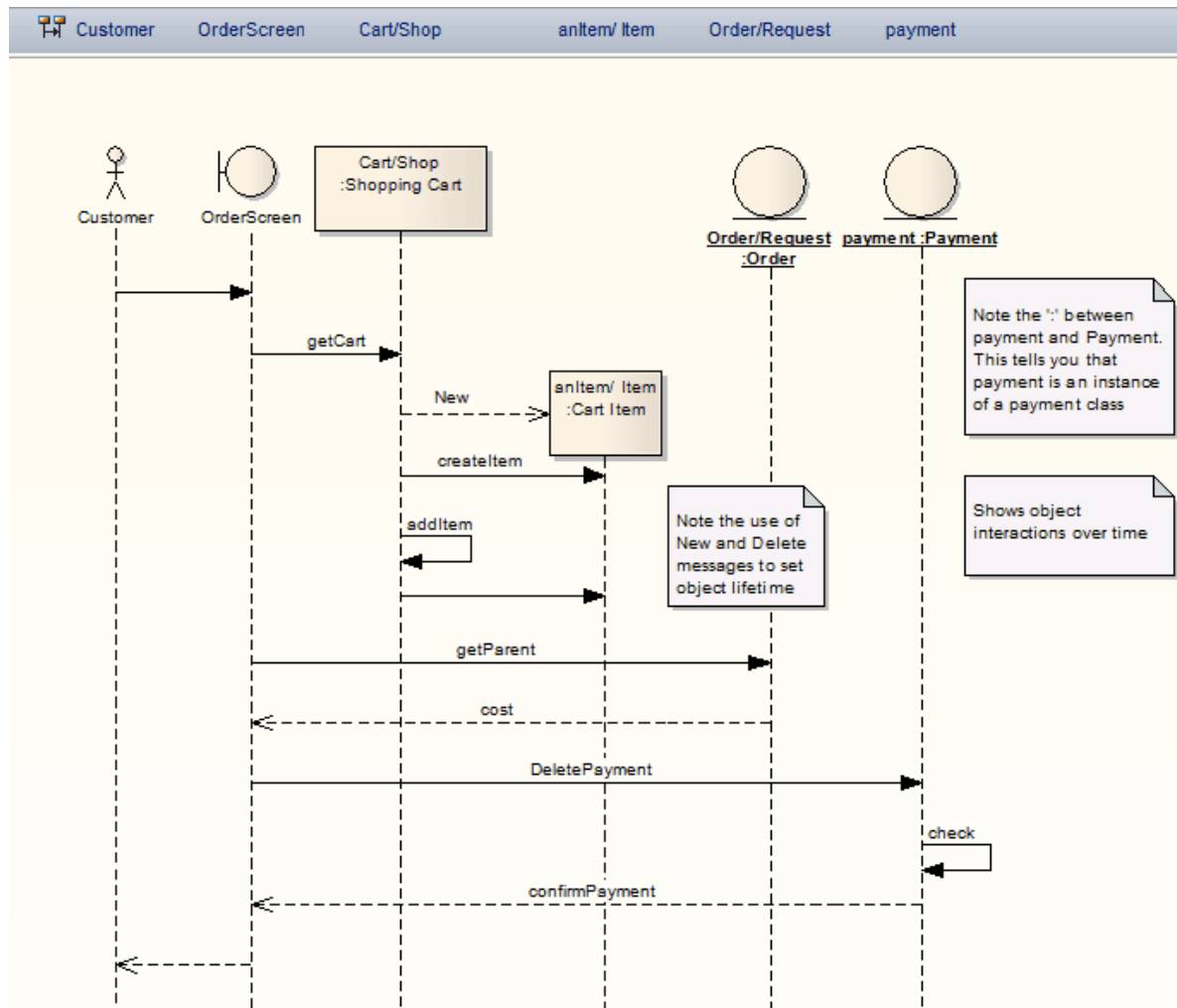
Display System Behavior

In displaying the behavior of a system as a Use Case diagram (see the *UML Dictionary*), Enterprise Architect gives the Business Analyst an easily understood tool for mapping the functional requirements and behavior of a system.

3.2 Software Architects

Software Architects can use Enterprise Architect to map functional requirements with Use Cases, perform real time modeling of objects using *Interaction diagrams* (Sequence, Timing, Communication or Interaction Overview), design the Deployment model and detail the deliverable components using Component diagrams.

For descriptions of the diagrams mentioned in this topic, see the *UML Dictionary*.



Map Functional Requirements of the System

With Enterprise Architect the Software Architect can take the high level business processes that have been modeled by the Business Analyst and create detailed Use Cases. Use Cases are used to describe the proposed functionality of a system and are only used to detail a single unit of discrete work.

Map Objects in Real Time

The Software Architect can use Interaction diagrams (Sequence and Communication diagrams) to model the dynamic design of the system. Sequence diagrams are used to detail the messages that are passed between objects and the lifetimes of the objects. Communication diagrams are similar to Sequence diagrams, but are used instead to display the way in which the object interacts with other objects.

Map Deployment of Objects

The Software Architect can use Deployment diagrams to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes. Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system.

Detail Deliverable Components

Component diagrams enable the Software Architect to model the physical aspects of a system. Components can be executables, libraries, data files or another physical resource that is part of a system. The component model can be developed from scratch from the Class model or can be brought in from existing projects and from third-party vendors.

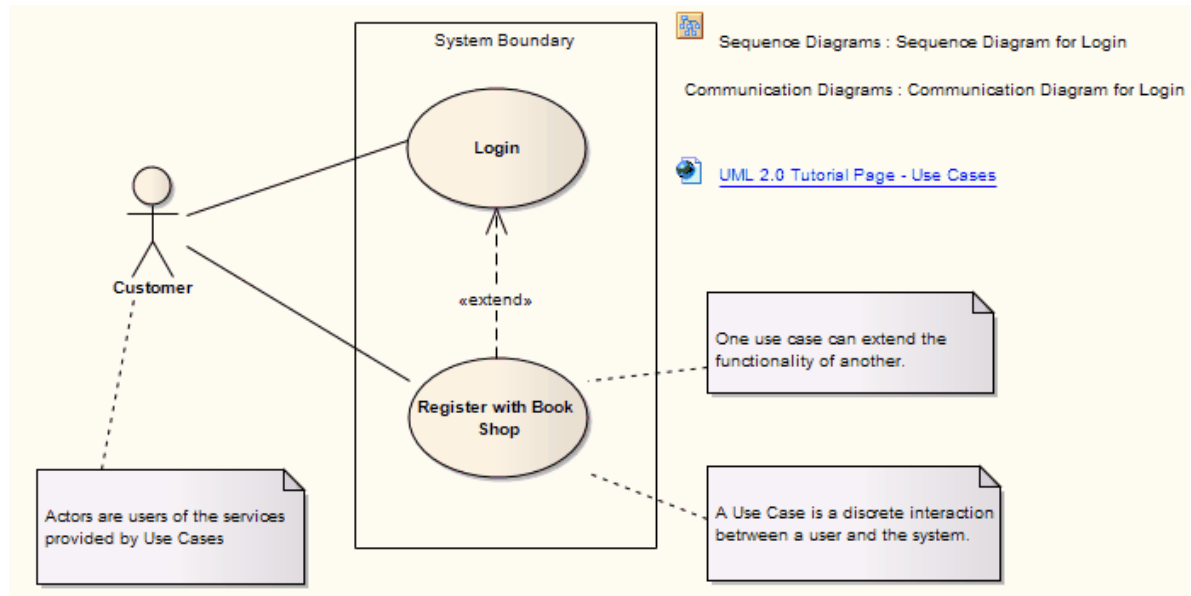
See Also

- Analysis Diagrams (in the *UML Dictionary*)
- *UML Modeling with Enterprise Architect - UML Modeling Tool*
- *XMI Import and Export* (in *UML Model Management*)
- *XML Technologies* (in *Code Engineering Using UML Models*)

3.3 Software Engineers

Software Engineers using Enterprise Architect can map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment with Deployment diagrams and define software packages with Package diagrams.

For descriptions of the diagrams mentioned in this topic, see the *UML Dictionary*.



Map Use Cases into Detailed Classes

In Enterprise Architect the Software Engineer can study the Use Cases developed by the Software Architect, and with that information create Classes that fulfill the objectives defined in the Use Cases. A Class is one of the standard UML constructs that is used to detail the pattern from which objects are produced at run time. To record the relationships between Use Cases and Classes, the Software Engineer can create diagrams linking the elements with Realization connectors, and/or map the Realization connectors in the Relationship Matrix (see *UML Modeling with Enterprise Architect – UML Modeling Tool*).

Detail Interaction Between Classes

Interaction diagrams (Sequence and Communication diagrams) enable the Software Engineer to model the dynamic design of the system. Sequence diagrams are used to detail the messages passed between objects and the lifetimes of the objects. Communication diagrams are similar to Sequence diagrams, but are used instead to display the way in which objects interact with other objects.

Define System Deployment

Deployment diagrams can be used to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes. Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system, to explain the connections and relationships of the components.

Define Software Packages

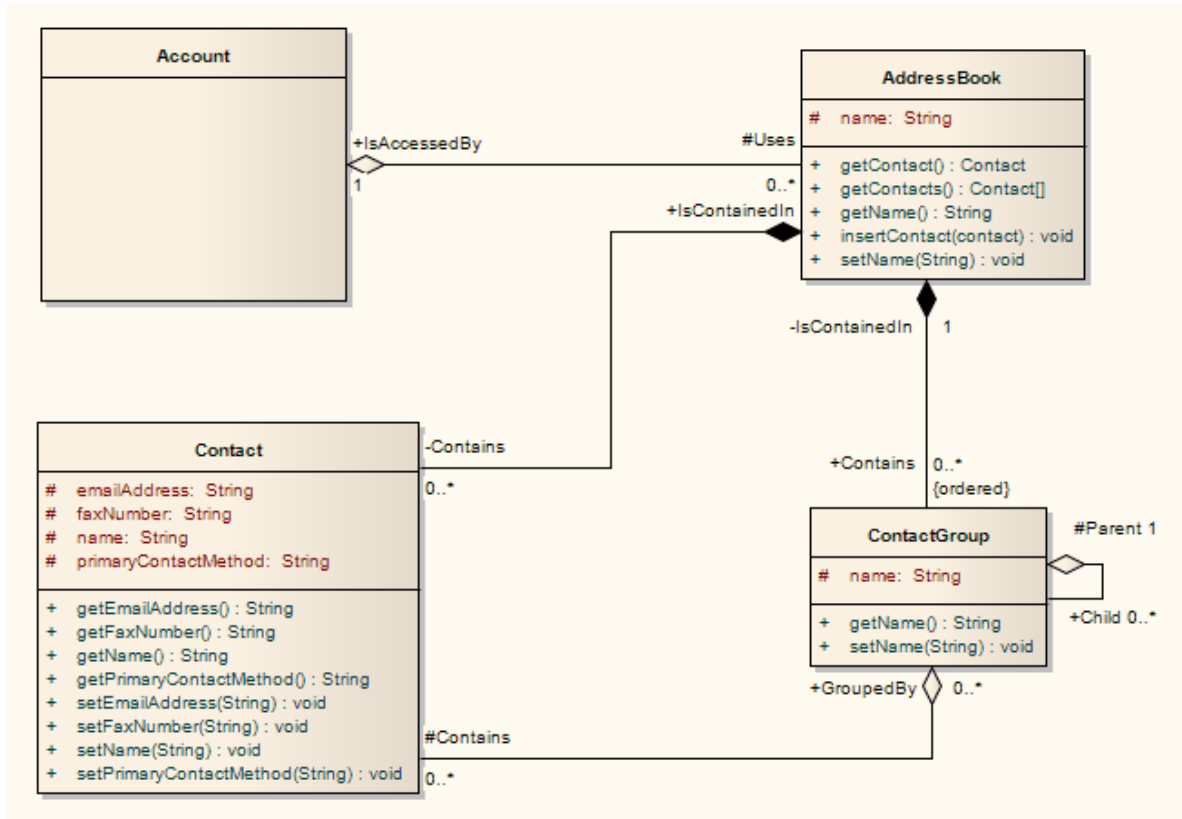
The Software Engineer can use Package diagrams to detail the software architecture. Package diagrams are used to organize diagrams and elements into manageable groups, declaring the dependencies.

See Also

- *UML Modeling with Enterprise Architect - UML Modeling Tool*

3.4 Developers

Developers can use Enterprise Architect to perform round trip code engineering, which includes reverse engineering of existing code and generation of code from UML Class elements. (See *Code Engineering Using UML Models*.)



State Machine, Package and Activity diagrams can be used by the developer to better understand the interaction between code elements and the arrangement of the code.

For descriptions of the diagrams mentioned in this topic, see the *UML Dictionary*.

Round Trip Engineering

Enterprise Architect gives the developer unparalleled flexibility, with the ability to round trip software from existing source code to UML 2.1 models and back again. Round trip Engineering involves both forward and reverse engineering of code. Keeping the model and code synchronized is an important aspect of round trip engineering. (See *Code Engineering Using UML Models*.)

Reverse Engineering

Enterprise Architect enables developers to reverse engineer code from a number of supported languages and view the existing code as Class diagrams. The developer can use Class diagrams to illustrate the static design view of the system. Class diagrams consist of Classes and interfaces and the relationships between them. The Classes defined in UML Class diagrams can have direct counterparts in the implementation of a programming language.

Forward Engineering

As well as the ability to reverse engineer code, Enterprise Architect offers the developer the option of forward engineering code (code generation). This enables the developer to make changes to their model with Enterprise Architect and have these changes implemented in the source code.

Determine the System State

To visualize the state of the system the developer can use State Machine diagrams to describe how elements

move between states, classifying their behavior according to transition triggers and constraining guards. State Machine diagrams are used to capture system changes over time, typically being associated with particular Classes (often a Class can have one or more State Machine diagrams used to fully describe its potential states).

Visualize Package Arrangement

Package diagrams are used to help design the architecture of the system. They are used to organize diagrams and elements into manageable groups, and to declare their dependencies.

Follow the Flow of Code

Activity diagrams are used to enable a better understanding of the flow of code. Activity diagrams illustrate the dynamic nature of the system. This enables modeling of the flow of control between Activities and represents the changes in state of the system.

See Also

- *XML Technologies (See Code Engineering Using UML Models)*
- *MDA Transformations User Guide*
- *Visual Execution Analyzer in Enterprise Architect*

3.5 Project Managers

Enterprise Architect provides support for the management of projects. Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort, and estimate project sizes. Enterprise Architect also helps them manage element status, change control and maintenance.

Provide Project Estimates

With Enterprise Architect the Project Manager has access to a comprehensive project estimation tool that calculates effort from Use Case and Actor objects, coupled with project configurations defining the technical and environmental complexity of the work environment.

Resource Management

Managing the allocation of resources in the design and development of system components is an important and difficult task. Enterprise Architect enables the Project Manager or Development Manager to assign resources directly to model elements and track progress over time.

Risk Management

The **Project Management** window can be used to assign Risk to an element within a project. The Risk Types enable the Project Manager to name the risk, define the type of risk, and give it a weighting.

Maintenance

Enterprise Architect enables the Project Manager to track and assign maintenance-related items to elements within Enterprise Architect. This enables rapid capture and record keeping for items such as issues, changes, defects and tasks. They can also create and maintain a project Glossary of processes, procedures, terms and descriptions.

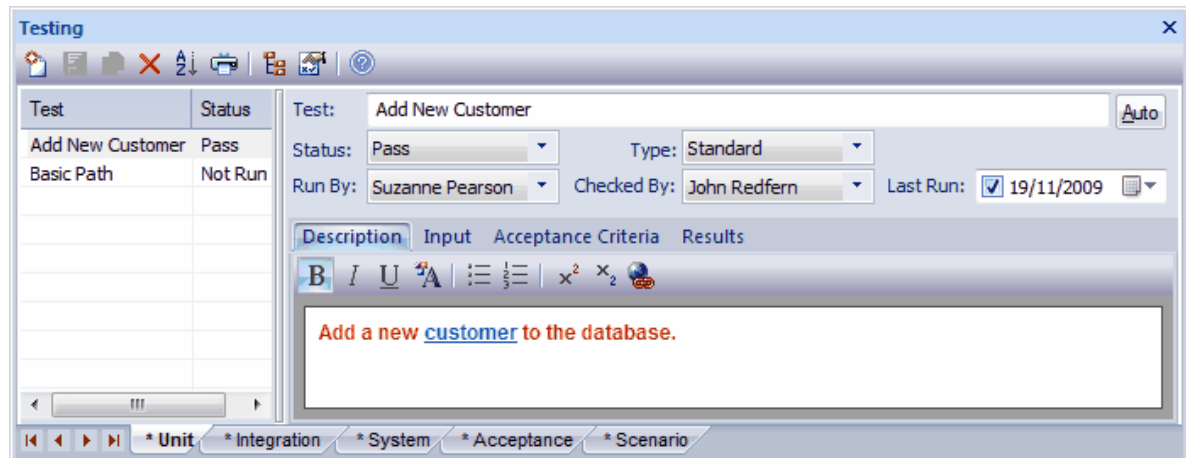
For further information, see *Project Management with Enterprise Architect*.

3.6 Testers

Enterprise Architect provides support for design testing by enabling you to create test scripts against elements in the modeling environment.

You assign test cases to individual model elements, requirements and constraints (see *UML Modeling with Enterprise Architect – UML Modeling Tool*). You can add scenarios to model elements, and use element defects to report problems associated with model elements (see *Project Management with Enterprise Architect*).

For more detailed information on testing, see the *Introduction to Testing in Enterprise Architect* topic in *Project Management with Enterprise Architect*.



Test Cases

With Enterprise Architect, Quality Assurance personnel can set a series of tests for each UML element. The test types include Unit testing, Acceptance testing, System testing and Scenario testing.

Import Requirements, Constraints and Scenarios

To help ensure that testing maintains integrity with the entire business process, Enterprise Architect enables the tester to import requirements, constraints and scenarios defined in earlier iterations of the development life cycle. Requirements indicate contractual obligations that elements must perform within the model. Constraints are conditions which must be met in order to pass the testing process. Constraints can be Pre-conditions (states which must be true before an event is processed), Post Conditions (events that must occur after the event is processed) or invariant constraints (which must remain true through the duration of the event). Scenarios are textual descriptions of an object's action over time and can be used to describe the way a test works.

Create Quality Test Documentation

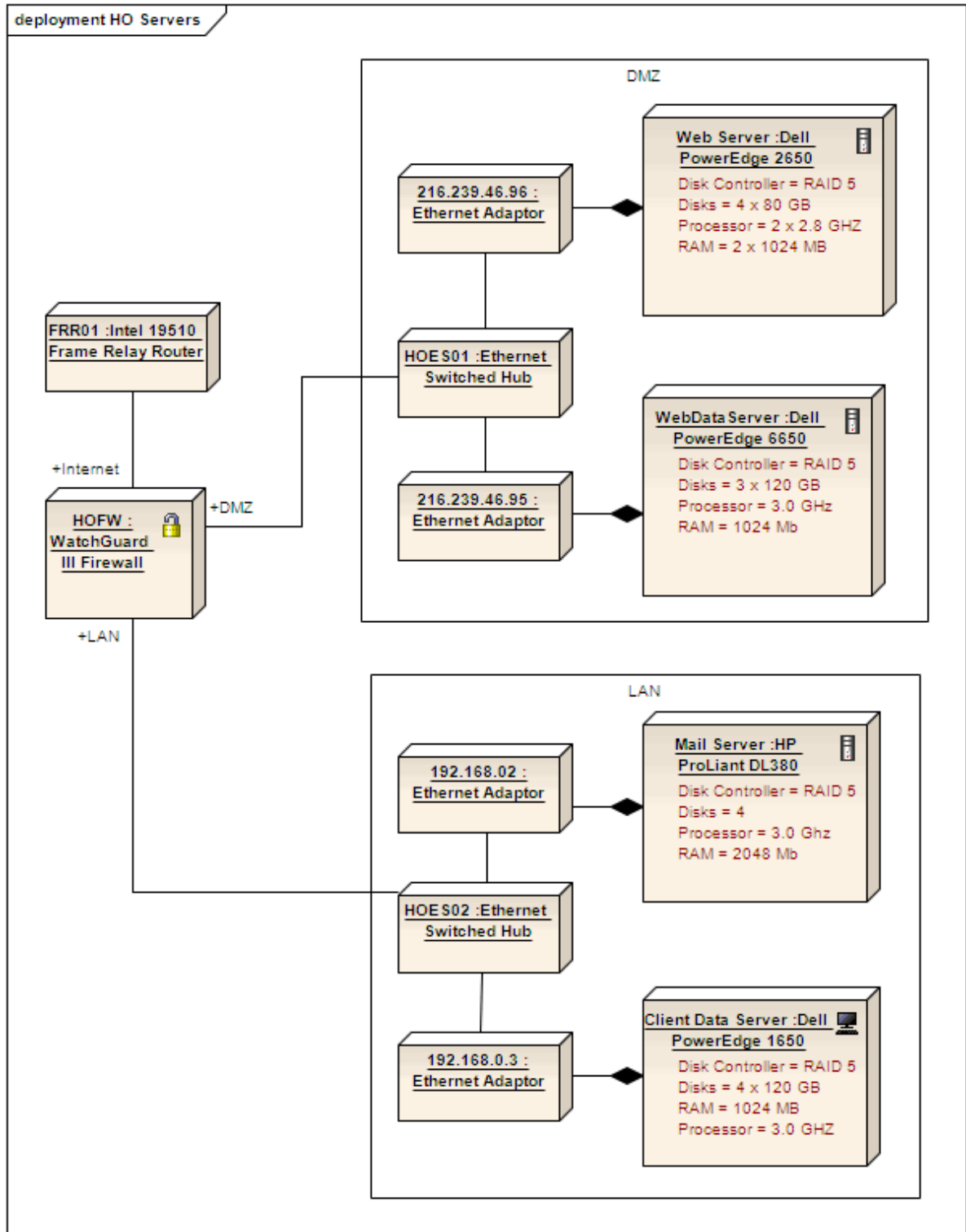
Enterprise Architect provides the facility to generate high quality test documentation. Enterprise Architect produces test documentation in the industry-standard .RTF file format.

Element Defect Changes

Defect tracking enables you to allocate defect reports to any element within the Enterprise Architect model. This enables all who are involved in the project to quickly view the status of defects, to see which defects have to be addressed and which have been dealt with.

3.7 Implementation Manager

Using Deployment diagrams (see the *UML Dictionary*) in Enterprise Architect, you can model the roll out of a project, including network deployment and workstation deployment. Users involved in project deployment can add maintenance tasks to the diagram elements.



Deployment diagrams provide a static view of the run-time configuration of nodes on the network or of workstations, and the components that run on the nodes or are used in the workstations.

Maintenance

Enterprise Architect enables you to track and assign maintenance-related items to elements within Enterprise Architect. (See *Project Management with Enterprise Architect*.) This enables you to rapidly capture and keep records of maintenance tasks such as issues, changes, defects and tasks. By providing a centralized facility for each element involved in the deployment process Enterprise Architect offers a powerful solution for tracing the maintenance of the items and processes involved in system deployment.

3.8 Technology Developers

Technology Developers are Enterprise Architect users who create customized additions to the functionality already present within Enterprise Architect. These additions include UML Profiles, UML Patterns, Code Templates, Tagged Value Types, Scripts, Custom Queries, Transformations, MDG Technologies and Enterprise Architect Add-Ins. By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.

UML Profiles

By creating UML Profiles the technology developer can create a customized extension for building UML models that are specific to a particular domain. Profiles are stored as XML files and can be imported into any model as required.

UML Patterns

Patterns are sets of collaborating Objects and Classes that provide a generic template for repeatable solutions to modeling problems. As patterns are discovered in any new project, the basic pattern template can be created. Patterns can be re-used with the appropriate variable names modified for any future project.

Code Templates

Code Templates are used to customize the output of source code generated by Enterprise Architect. This enables the generation of code languages not specifically supported by Enterprise Architect and enables you to define the way Enterprise Architect generates source code to comply with your own company style guidelines.

Tagged Value Types

Tagged Values are used in Enterprise Architect to extend the information relating to an element in addition to the information directly supported by the UML language. A Tagged Value, strictly, is the *value* of a property of a modeling item, the property being called a *tag*. For example, a Class element called *Person* might have a tag called **Age** with the Tagged Value of **42**. More loosely, the combination of tag and value can be referred to as a Tagged Value.

A *Tagged Value Type* is a group of parameters that define and/or limit the possible values of a tag and, in many instances, how a specific value is assigned to the tag. For example, the tag **Age** might have a Tagged Value Type of **Integer**, so the user simply types in a numeric value. Alternatively, the type could be **Spin**, with lower and upper limits of, say, **20** and **120**, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120.

Typically, Tagged Values are used during the code generation process, or by other tools to pass on information that is used to operate on elements in particular ways.

MDG Technologies

MDG Technologies can be used to create a logical collection of resources that can contain UML Profiles, Patterns, Code Templates, Image files and Tagged Value types that are accessed through a technology file.

Enterprise Architect Add-Ins

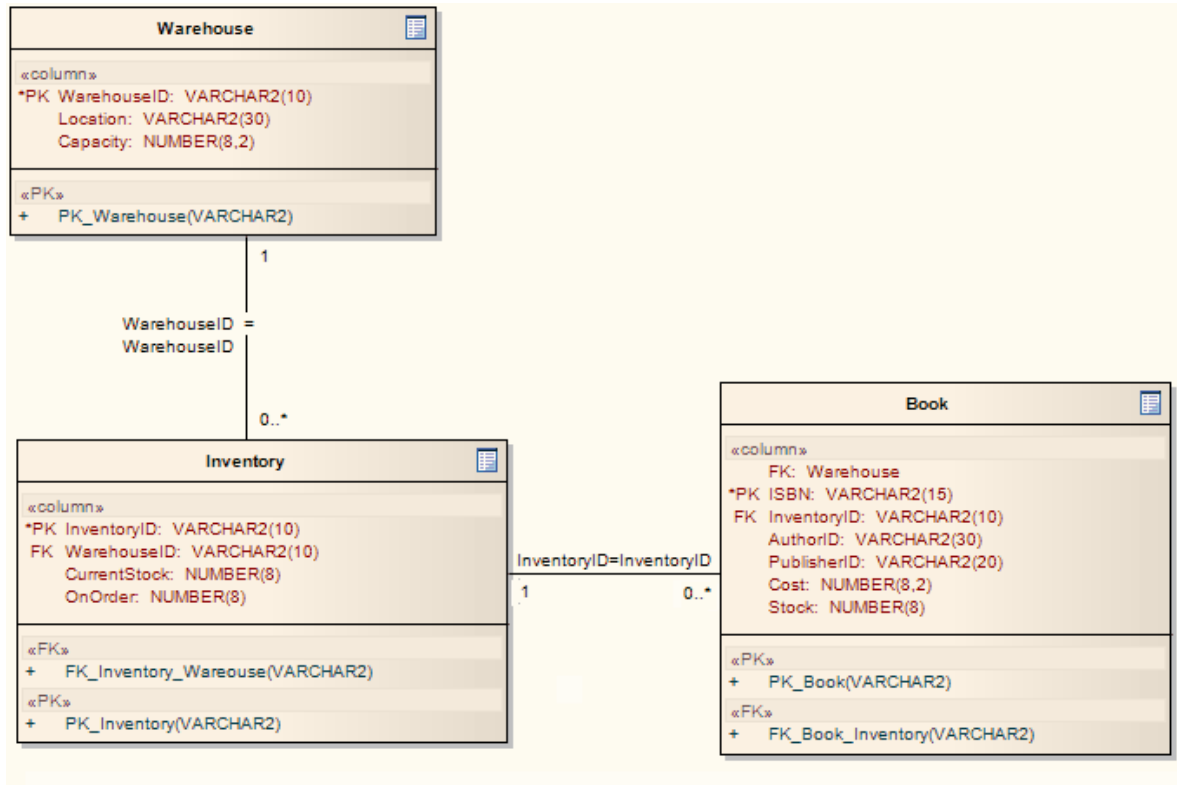
Enterprise Architect Add-Ins enable you to build your own functionality into Enterprise Architect, creating your own mini programs that can extend the capabilities of Enterprise Architect, defining your own menus, and creating your own Custom Views.

The Enterprise Architect Software Developers' Kit

For explanations of how the features identified above are used within Enterprise Architect, see *Extending UML With Enterprise Architect*. However, *SDK for Enterprise Architect* explains in detail how to *develop and integrate* these facilities.

3.9 Database Developers

Enterprise Architect supports a range of features for the development of databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model. (See *Code Engineering Using UML Models*.)



Create Logical Data Models

With Enterprise Architect the Database developer can build database diagrams using the built-in UML Data Modeling Profile. This supports the definition of Primary and Foreign keys, cardinality, validation, triggers, constraints and indexes.

Generate Schema

By using Enterprise Architect's DDL generation function the Database Administrator can create a DDL script for creation of the database table structure from the model. Enterprise Architect currently supports JET-based databases; DB2; InterBase; Informix; Ingres; MySQL; SQL Server; PostgreSQL; Sybase Adaptive Server Anywhere and Adaptive Server Enterprise; and Oracle 9i, 10g and 11g.

Reverse Engineer Database

Using an ODBC data connection the Database Administrator can import a database structure from an existing database to create a model of the database. Generating the model directly from the database enables the DBA to quickly document their work and create a diagrammatic account of a complex database through the graphical benefits of UML.

See Also

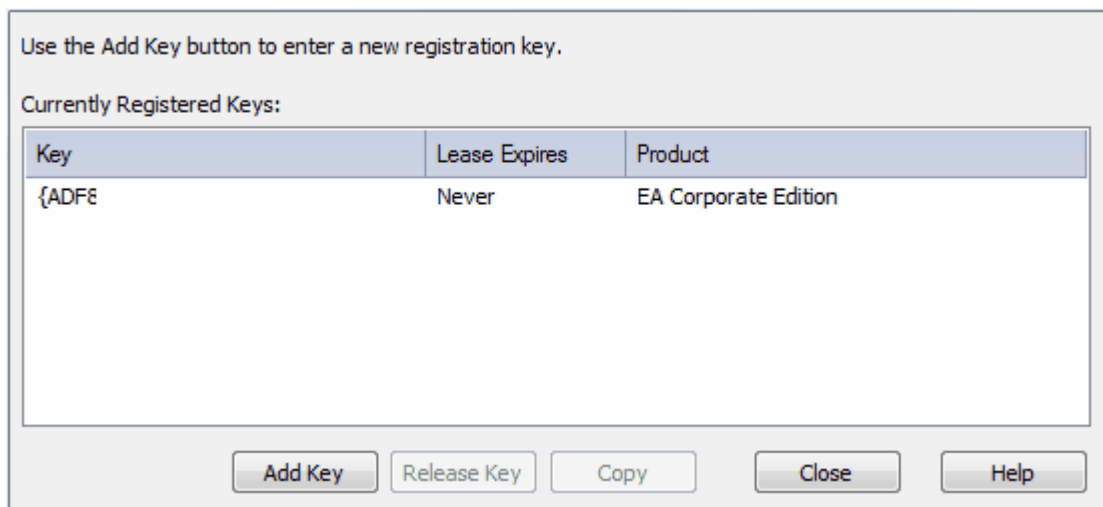
- *Set Up a Database Repository, in UML Model Management*

4 License Management



The **License Management** dialog in Enterprise Architect enables you to upgrade Enterprise Architect and to register Add-Ins.

To access License Management from within Enterprise Architect, select the **Help | Register and Manage License Key(s)** menu option. The **License Management** dialog displays, listing the currently-registered keys, when the key expires (or, for shared keys, when they are to be reactivated in the keystore for issue to another user) and the product each key applies to.



Use the buttons on the dialog as required:

Option	Use to
Add Key	Display the Add Registration Key dialog, which enables you to: <ul style="list-style-type: none"> • Add a new key to update to a higher version of Enterprise Architect or to register an Add-In. • Obtain a key from the Sparx Systems Key Store (available for version 4.51 and above). For more information on adding keys see the Add License Key ^[56] topic.
Remove Key	Make the Add-In or current version of Enterprise Architect inoperable.
Copy	Place the highlighted key into the clipboard.
Close	Close the dialog.
Help	Display the help for this topic.

You can also run the following tasks from the **License Management** dialog:

- [Register a Full License](#)^[26]
- [Upgrade an Existing License](#)^[59]
- [Register an Add-In](#)^[62]

4.1 Finding Your License Information

You can find information on your Enterprise Architect license in the **About Enterprise Architect** dialog; select the **Help | About EA** menu option.



4.2 Add License Key

Two types of key can be used in conjunction with Enterprise Architect:

- Private keys allow you to register, indefinitely, an Enterprise Architect license (Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering or Ultimate) or an Add-In product (such as MDG Link for Eclipse or MDG Link for Visual Studio.NET) on the machine and user account that you are currently using.
- Shared keys - obtained from a central shared key store - give you temporary access to the installed 'suite' edition of Enterprise Architect. Shared Keys are available with the purchase of a floating license, and require Enterprise Architect version 4.51 or higher. For more information, see [Enterprise Architect Floating License](#).

Notes:

- Some license keys can override and remove others. For example, the key for a more advanced edition of Enterprise Architect replaces the key for a simpler edition, and the key for MDG Integration for Visual Studio replaces the key for MDG Link for Visual Studio.
- Shared keys and Private keys have different formats and cannot be used interchangeably in the two procedures below.

Add a Private Key

To add a private key, follow the steps below:

1. Select the **Help | Register and Manage License Key(s)** menu option. The [License Management](#) ⁵⁴ dialog displays.
2. Click on the **Add Key** button. The **Add Registration Key** dialog displays.

3. Click on the **Enter Private Key** tab.
4. In the **Name** and **Company** fields, type your user name and company name. Into the registration key field, copy the registration key (including any parentheses around the key).
5. Click on the **OK** button to confirm the key selection.


Add a Shared Key

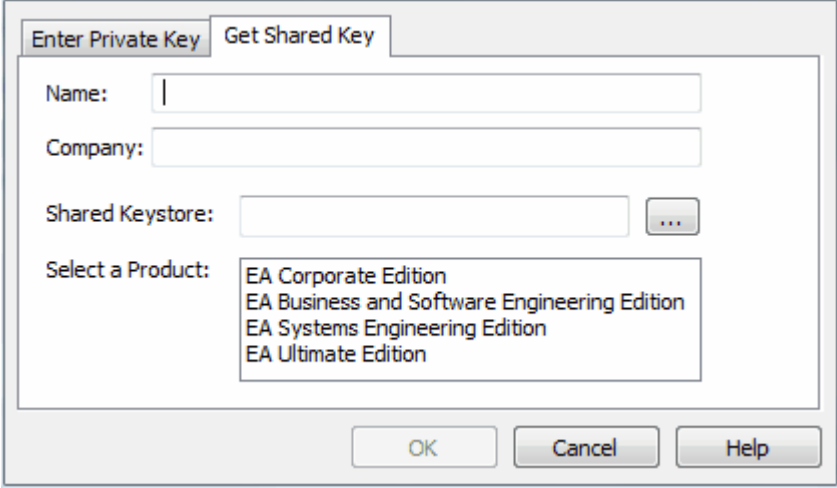
Shared Keys require a shared license keystore to be configured by your license administrator. The key store can be either file-based or network based (preferred). Only the Key Administrator is required to install the Sparx Enterprise Key Store application. End users simply connect to the configured key file - advised by the administrator - using Enterprise Architect as described below. No additional software is required to be installed.

Note:

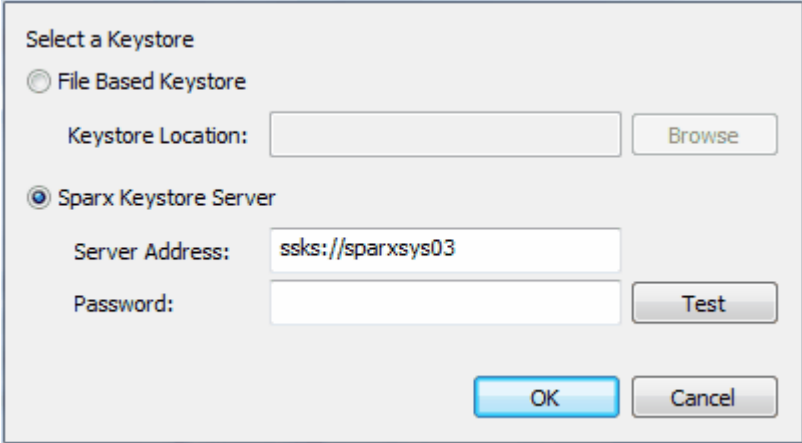
If any error messages are displayed while attempting to obtain a shared key, see [Keystore Troubleshooting](#) ⁵⁸.

To add a shared key, follow the steps below:

1. Select the **Help | Register and Manage License Key(s)** menu option. The [License Management](#)  dialog displays.
2. Click on the **Add Key** button. The **Add Registration Key** dialog displays.
3. Click on the **Get Shared Key** tab.



4. In the **Name** and **Company** fields, type your user name and company name.
5. In the **Shared key store** field, click on the [...] (Browse) button. The **Shared Keystore Selection** dialog displays.



6. If your keystore is file-based, select the **File Based Keystore** radio button, click on the **Browse** button, and locate and select the keystore file. Go to step 9.
7. If the keystore is network-based, click on the **Sparx Keystore Server** radio button and, in the **Server Address** field, type the server address of the keystore.
8. If necessary, type in the password (advised by your administrator) and/or click on the **Test** button to ensure that you have a connection to the keystore.
9. Click on the **OK** button to return to the **Get Shared Key** tab, which now shows the name of the keystore in the **Shared Keystore** field.
10. In the **Select a Product** field, click on the appropriate product name.
11. Click on the **OK** button. The **License Management** dialog redisplay, indicating that the shared key is registered for the selected product, until the key expiry date.
12. Click on the **Close** button.

4.3 Keystore Troubleshooting

Message Displayed:	Explanation
<p><i>Error reading Key Store file: (Access is denied)</i></p>	<p>All users who are to use the shared key facility require Read, Write and Modify access to the <i>sskeys.dat</i> file containing the shared keys. Please verify that all required users have sufficient permissions to the file and try again.</p> <p>If the problem continues, contact Sparx Support.</p> <p>Tip:</p> <p>Review the effective permissions calculated at the location of the key file for the user account reporting the problem. You should closely examine the permissions for both the Network Share and the File System. It is possible that these permissions have been overwritten at some point.</p>
<p><i>Error reading Key Store file: (Key File has been moved)</i></p>	<p>As a security measure in the key store, the hard drive serial number is recorded when the file is created. The file then cannot be moved from the original location in which it was created. If the key store has to be re-located for any reason, the administrator should re-create the key store in the new location using the original license keys.</p> <p>This issue is commonly seen after a file server has undergone a hardware upgrade in which the physical hard drives have been replaced. Problems could also occur if the drive used is part of a RAID configuration.</p> <p>This message could also appear where the key store exists on a Novell-based file system. When creating the key store, the administrator is prompted to confirm that the key store is to be located on a Novell Netware file server. If the administrator clicks on the Yes button, the key store instead records the logical path used to create it, and all users must connect to the key store using this same path. The recorded path is case-sensitive and must be an exact match.</p>

4.4 Upgrade an Existing License

Enterprise Architect comes in six editions: Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate. If you are using a less powerful edition, such as Professional or Desktop, you can upgrade your license at a future date. You can do this by purchasing an upgrade key from Sparx Systems (see the Sparx Systems [website](#) for purchase details).

An upgrade key is a special key that upgrades an existing license to a higher *edition*. Once you have purchased and received the appropriate key, use the following procedures to unlock additional features. The procedure for Enterprise Architect version [7.0 and later](#) releases differs from the procedure for [earlier releases](#).

Notes:

- The [Lite](#) version and the Trial version cannot be registered or upgraded. If you have purchased Enterprise Architect, you must download the registered version from www.sparxsystems.com/securedownloads/easetupfull.exe before you can enter your registration key.
- This topic is mainly relevant to users with *private* keys. If you are an end-user with a *shared* key, you would simply be allocated the relevant key next time you requested one. If, however, you need to upgrade while using a shared key on a long lease, you would simply click on the [Release Key](#) button and then the [Add Key](#) button.

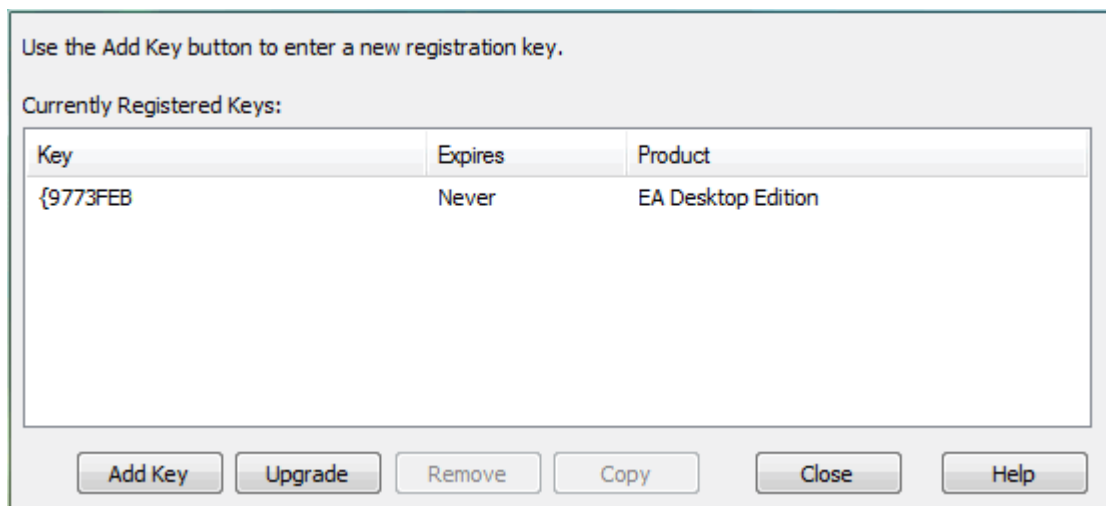
Tip:

Once you have successfully completed an upgrade with a *Private* key, select the **Help | About EA** menu option. Copy the registration key shown and store it somewhere safe; this is a key to the full license of the edition you have upgraded to. If you ever have to reinstall Enterprise Architect, you can register it with this key, so you won't have to go through the upgrade process again.

Upgrade Enterprise Architect Version 6.5 and Earlier

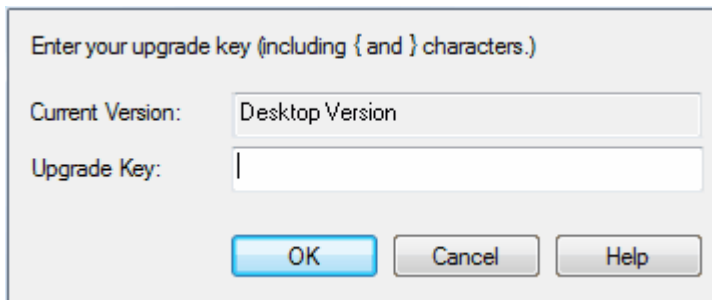
To upgrade from one license edition to another, follow the steps below:

1. Make sure you have a valid upgrade key purchased from Sparx Systems; you typically receive this in an email or PDF format.
2. Open Enterprise Architect.
3. Select the **Help | Register and Manage License Key(s)** menu option. The **License Management** dialog displays



4. Click on the **Add Key** button or the **Upgrade** button to enter a new license key.
5. If you selected the **Add key** option, the **Add Registration Key** dialog displays. Enter the key you received for the upgraded edition of Enterprise Architect, including the { and } bracket characters (use copy and paste from an email to avoid typing mistakes).
6. If you selected the **Upgrade** option, the **Upgrade Key** dialog displays. Enter the key you received for the upgraded edition of Enterprise Architect, including the { and } bracket characters (use copy and paste

from an email to avoid typing mistakes).



Enter your upgrade key (including { and } characters.)

Current Version: Desktop Version

Upgrade Key: |

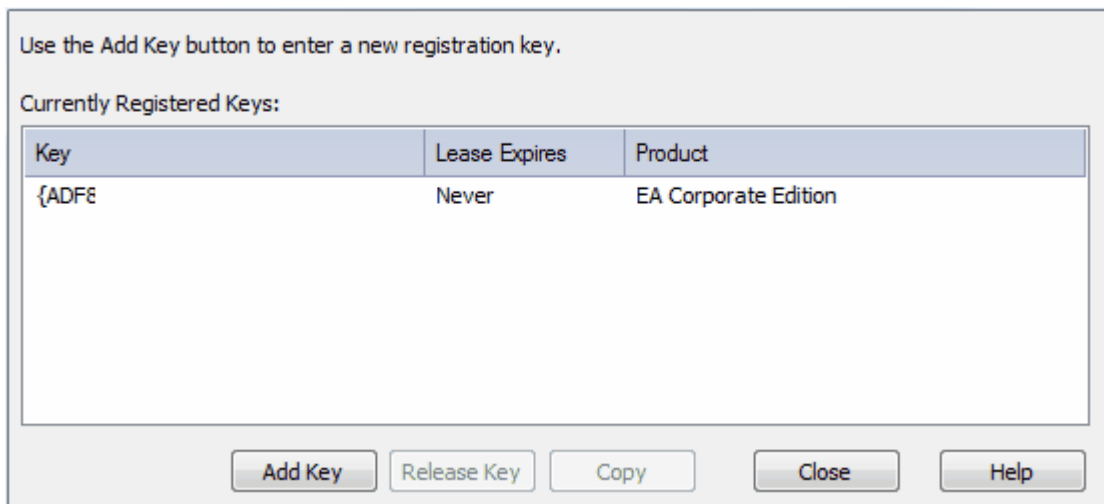
OK Cancel Help

7. Click on the **OK** button. If the key is valid, Enterprise Architect modifies the **Current Version** field to reflect the upgrade.
8. Close Enterprise Architect and restart to enable the unlocked features.

Upgrade Enterprise Architect Version 7.0 and Later

To upgrade from one license edition to another, follow the steps below:

1. Make sure you have a valid upgrade key purchased from Sparx Systems; you typically receive this in an email or PDF format.
2. Open Enterprise Architect.
3. Select the **Help | Register and Manage License Key(s)** menu option. The **License Management** dialog displays.



Use the Add Key button to enter a new registration key.

Currently Registered Keys:

Key	Lease Expires	Product
{ADFE	Never	EA Corporate Edition

Add Key Release Key Copy Close Help

4. Click on the **Add Key** button; the **Add Registration Key** dialog displays

Enter Private Key | Get Shared Key

Name:

Company:

Copy registration key into space below, then press OK button

OK Cancel Help

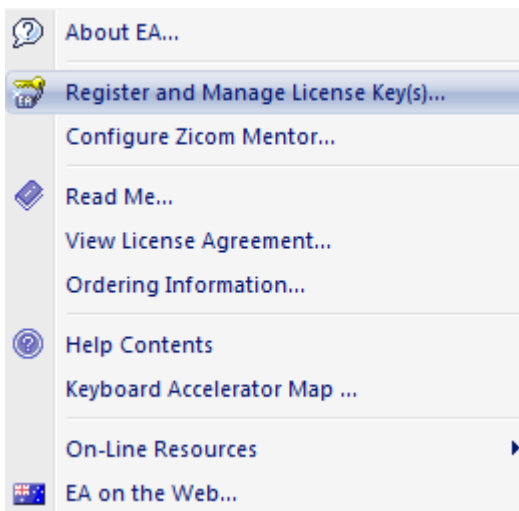
5. In the **Name** and **Company** fields, type your name and company name.
6. To avoid typing mistakes, copy the key you received for the upgraded edition of Enterprise Architect - including the { and } bracket characters - from the email and paste the key into the **Copy registration key** field.
7. Click on the **OK** button. Enterprise Architect displays a *Registration succeeded – Thank you for purchasing Enterprise Architect xxxx Edition* message.
8. Click on the **OK** button, and then on the **Close** button to continue working in Enterprise Architect.

4.5 Register Add-In

To register Add-Ins for Enterprise Architect, follow the steps below:

Register an Add-In for Enterprise Architect

1. Purchase one or more licenses for the Add-In from your Add-In provider. Once you have paid for a licensed version of the Add-In, you receive (via email or other suitable means) a license key for the product.
2. Save the license key and the latest full version of the Add-In.
3. Run the Add-In's setup program to install the Add-In.
4. In Enterprise Architect, select the **Help | Register and Manage License Key(s)** menu option, or the **Add-Ins | Enter License Key for <Add-In name>** menu option.



The [License Management](#) ⁵⁴ dialog displays.

5. Click on the **Add Key** button. The **Add Registration Key** dialog displays.

6. Type in the key you received with the Add-In, including the { and } characters.

Note:

If you have purchased a floating licence (shared key) follow [these steps](#) ⁵⁷.

7. Click on the **OK** button.
8. When the Add-In has been added successfully, close down Enterprise Architect and restart it to apply the changes.

Index

- A -

- Acknowledgement
 - CXImage Library 22
 - Of Contributions 22
 - Of Trademarks 21
 - Print Listview 22
- Add
 - Connectors To UML Model, Quick Start 32
 - Diagram To Package 31
 - Diagram To UML Model, Quick Start 31
 - Element To Diagram 31
 - Element To UML Model, Quick Start 31
 - Key 56
 - License Key 56
 - Package To UML Model, Quick Start 31
- Add-In
 - MDG 15
 - Model Driven Generation 15
 - Register 62

- B -

- Bend Connector 32
- Build Systems Using UML
 - Enterprise Architect 4
- Business Analyst
 - And Enterprise Architect 41
 - Project Role 41
- Business and Software Engineering Edition
 - Of Enterprise Architect 12

- C -

- CASE Tool
 - Enterprise Architect 2
- Code Engineering
 - Broad View Of, In Enterprise Architect 11
 - With Enterprise Architect 11
- Community Site 16
- Compiled April 13 2010 2
- Complex Modeling
 - Enterprise Architect 4
- Connector
 - Add To Diagram 32
 - Add To UML Model, Quick Start 32
 - Bend Connector 32

- Move 35
- Copy
 - Elements Between Diagrams 35
- Copyright Notice 18
- CORBA
 - MDG Technology For, Enterprise Architect 15
- Corporate Edition
 - Of Enterprise Architect 12
- Create
 - A Project In Enterprise Architect 29
- Cut And Paste
 - Elements Between Diagrams 35

- D -

- Data Distribution Service
 - MDG Technology For, Enterprise Architect 15
- Database Administrator
 - And Enterprise Architect 53
 - Project Role 53
- Database Modeling
 - Enterprise Architect 4
- DDS
 - MDG Technology For, Enterprise Architect 15
- Debug
 - Enterprise Architect 4
- Delete
 - Connectors, Quick Start 36
 - Diagrams, Quick Start 36
 - Elements, Quick Start 36
 - Packages, Quick Start 36
- Deployment and Rollout
 - And Enterprise Architect 50
 - Project Role 50
- Design Systems Using UML
 - Enterprise Architect 4
- Desktop Edition
 - Of Enterprise Architect 12
- Developer
 - And Enterprise Architect 46
 - Forward Engineering 46
 - Project Role 46
 - Reverse Engineering 46
 - Round-Trip Engineering 46
 - Visualise Package Arrangement 46
- Diagram
 - Add To UML Model, Quick Start 31
 - Diagram To Package 31
 - Move, Impact On Element 35
- Document Generation
 - Enterprise Architect 4
- DoDAF-MODAF

- DoDAF-MODAF
 - MDG Technology For, Enterprise Architect 15
- DOORS, Telelogic
 - MDG Link For, Enterprise Architect 15
- E -**
- Eclipse
 - MDG Integration For, Enterprise Architect 15
 - MDG Link For, Enterprise Architect 15
- Editions
 - Business and Software Engineering 12
 - Corporate 12
 - Desktop 12
 - Floating Licence 12
 - Lite 14
 - Of Enterprise Architect 12
 - Professional 12
 - Standalone 12
 - System Engineering 12
 - Ultimate 12
- Element
 - Add To Diagram 31
 - Add To UML Model, Quick Start 31
 - Copy And Paste Between Diagrams 35
 - Cut And Paste Between Diagrams 35
 - Move, Impact Of Diagram 35
- End User License Agreement 18
- Enterprise Architect
 - And Deployment 50
 - And IDEs 4
 - Build Systems 4
 - CASE Tool 2
 - Code Engineering With 11
 - Community Site 16
 - Create Project, Tutorial 29
 - Database Modeling 4
 - Debug 4
 - Design Systems 4
 - Editions, Differences Between 12
 - End User License Agreement 18
 - Extend Trial Period 23
 - Feedback Pages 2
 - For Business Analysts 41
 - For Database Administrators 53
 - For Developers 46
 - For Implementation Managers 50
 - For Project Managers 48
 - For Software Architects 43
 - For Software Engineers 45
 - For Technology Developers 52
 - For Testers 49
 - Formal Statements 18
 - Fundamental Processes 8
 - Generate Documentation 4
 - Generate Source Code 4
 - Help 16
 - How To Use 2
 - Install 25
 - Introduction 2
 - Key Features 7
 - License Agreement 18
 - Lite Edition 14
 - Manage Model Structure 4
 - Manage Requirements 4
 - MDA Transformation 4
 - Model Complexity 4
 - Model Requirements 4
 - Modeling Tool 2
 - Online User Guide 2
 - Order 25
 - Pricing And Purchasing 25
 - Professional Roles 40
 - Project Roles 40
 - Quick Start 28
 - Read-Only Edition 14
 - Register Full License 26
 - Reverse Engineer 4
 - Select Edition To Trial 23
 - Share Model Development 4
 - Software Product License Agreement 18
 - Support 16, 17
 - Trial Version 23
 - Tutorial, Modeling 28
 - UML 2.1 Support 4
 - Uses Of 4
 - Visualize Systems 4
 - What Can I Do With It? 4
 - What is Enterprise Architect? 4
 - Working With 8, 28
- Enterprise Architect Lite
 - Availability 14
 - Description 14
 - Edition 14
- Enterprise Java Beans
 - MDG Technology For, Enterprise Architect 15
- F -**
- Features
 - Of Enterprise Architect 7
- Feedback Pages 2
- Floating Licence
 - Version of Corporate Edition 12

Formal Statements 18

- G -

Generate Source Code

Enterprise Architect 4

Getting Started

Add License Key 56

Installing Enterprise Architect 25

License Information 55

License Management 54

Quick Start Guide To Enterprise Architect 28

Register Enterprise Architect 26

Registration Key 55

Upgrade Existing License 59

- H -

Help

File Formats 17

Index Tab 16

Release Date 17

Search Tab 16

Systems 16

Version Details 17

- I -

IDE

And Enterprise Architect 4

Implementation Manager

And Enterprise Architect 50

Project Role 50

Install

Enterprise Architect 25

Introduction

Copyright Notice 18

Enterprise Architect Key Features 7

Help File Formats 17

License Agreement 18

Order Enterprise Architect 25

Support 17

To Enterprise Architect 2, 4

Trademarks 21

- K -

Key Features

Of Enterprise Architect 7

Key Store

File Based 56

Network Based 56

Keystore

Troubleshooting 58

- L -

License

Agreement 18

Information 55

Management 54

- M -

Manage

Models, Enterprise Architect Tasks 10

MDA Transformation

Enterprise Architect 4

MDG Add-In For

CORBA 15

Data Distribution Service 15

DDS 15

Department Of Defense Architecture Framework
- Ministry Of Defence Architecture Framework
15

DoDAF-MODAF 15

DOORS 15

Eclipse 15

Enterprise Java Beans 15

Python 15

SysML 15

Systems Modeling Languages 15

TcSE 15

Teamcenter Systems Engineering, Siemens PLM
15

Testing 15

The Open Group Architecture Framework 15

TOGAF 15

Visio 15

Visual Studio 2005/2008 15

Visual Studio.NET 15

Zachman Framework 15

MDG Integration For

Eclipse 15

TcSE 15

Teamcenter Systems Engineering, Siemens PLM
15

Visual Studio 2005/2008 15

MDG Link For

DOORS, Telelogic 15

Eclipse 15

Visio 15

Visual Studio.NET 15

MDG Technology For

- MDG Technology For
 - CORBA 15
 - Data Distribution Service 15
 - DDS 15
 - Department Of Defense Architecture Framework - Ministry Of Defence Architecture Framework 15
 - DoDAF-MODAF 15
 - Enterprise Java Beans 15
 - Python 15
 - SysML 15
 - Systems Modeling Languages 15
 - Testing 15
 - The Open Group Architecture Framework 15
 - TOGAF 15
 - Zachman Framework 15
 - Model
 - Manage In Enterprise Architect 10
 - Model Driven Generation
 - Add-Ins 15
 - Model Pattern
 - Select In Enterprise Architect 29
 - Model Wizard
 - Quick Start 29
 - Modeling
 - Enterprise Architect Processes 8
 - Enterprise Architect, Tutorial 28
 - Quick Start - Enterprise Architect 28
 - Tasks With Enterprise Architect 8
 - Tutorial - Enterprise Architect 28
 - With Enterprise Architect, Overview 9
 - Working With Enterprise Architect 28
 - Modeling Tool
 - Enterprise Architect 2
 - Move
 - Connectors, Quick Start 35
 - Diagrams, Quick Start 35
 - Elements, Quick Start 35
 - Packages, Quick Start 35
- O -**
- Option
 - Read-Only, For Enterprise Architect 14
 - Order
 - Enterprise Architect 25
- P -**
- Package
 - Add To UML Model, Quick Start 31
 - Move 35
 - Private Key
 - Add 56
 - Professional Edition
 - Of Enterprise Architect 12
 - Project
 - Create In Enterprise Architect, Tutorial 29
 - Tasks 38
 - Project Manager
 - And Enterprise Architect 48
 - Project Estimation 48
 - Project Role 48
 - Resource Management 48
 - Risk Management 48
 - Project Role
 - And Enterprise Architect 40
 - Business Analyst 41
 - Database Administrator 53
 - Deployment and Rollout 50
 - Developer 46
 - Implementation Manager 50
 - Project Manager 48
 - Software Architect 43
 - Software Engineer 45
 - Technology Developer 52
 - Tester 49
 - Purchase
 - Enterprise Architect 25
 - Python
 - MDG Technology For, Enterprise Architect 15
- Q -**
- Quick Start
 - Add Connectors To UML Model 32
 - Add Diagram To Package 31
 - Add Diagram To UML Model 31
 - Add Element To Diagram 31
 - Add Element To UML Model 31
 - Add Package to UML Model 31
 - Create A Project In Enterprise Architect 29
 - Define Connector Properties 33
 - Define Element Properties 33
 - Delete UML Model Components 36
 - Guide To Enterprise Architect 28
 - Move Project Components 35
 - Project Tasks 38
 - Save Project Changes 37
- R -**
- Read-Only Options
 - For Enterprise Architect 14

- Register
 - Add-Ins 62
 - Enterprise Architect 26
 - Registration Key
 - In License Information 55
 - Requirements Management
 - Enterprise Architect 4
 - Requirements Modeling
 - Enterprise Architect 4
 - Reset Sort Order
 - In Project Browser 35
 - Reverse Engineer
 - Enterprise Architect 4
- S -**
- Save
 - Changes 37
 - Diagram, Quick Start 37
 - Service Oriented Architecture
 - Development 11
 - Implementing XML-Based, In Enterprise Architect 11
 - Shared Key
 - Add 56
 - Issues 58
 - Shared Model Development
 - Enterprise Architect 4
 - SOA
 - Development 11
 - Implementing XML-Based, In Enterprise Architect 11
 - Software Architect
 - And Enterprise Architect 43
 - Project Role 43
 - Software Engineer
 - And Enterprise Architect 45
 - Project Role 45
 - Software Product License Agreement 18
 - Sort Order
 - Project Browser, Reset 35
 - Sparx Systems
 - Enterprise Architect Community Site 16
 - Website 16
 - Start Page
 - Quick Start 29
 - Support
 - For Registered Users 17
 - For Trial Users 17
 - SysML
 - MDG Technology For, Enterprise Architect 15
 - Systems Engineering Edition
 - Of Enterprise Architect 12
 - Systems Modeling Languages
 - MDG Technology For, Enterprise Architect 15
- T -**
- Tasks Pane
 - Introduction 16
 - TcSE
 - MDG Integration For, Enterprise Architect 15
 - Teamcenter Systems Engineering, Siemens PLM
 - MDG Integration For, Enterprise Architect 15
 - Technology Developer
 - And Enterprise Architect 52
 - Project Role 52
 - Tester
 - And Enterprise Architect 49
 - Project Role 49
 - Testing
 - MDG Technology For, Enterprise Architect 15
 - The Open Group Architecture Framework
 - MDG Technology For, Enterprise Architect 15
 - TOGAF
 - MDG Technology For, Enterprise Architect 15
 - Trademarks 21
 - Trial Version
 - Extend Trial Period 23
 - Of Enterprise Architect 23
 - Select Edition To Trial 23
 - Select Workspace Layout 23
- U -**
- Ultimate Edition
 - Of Enterprise Architect 12
 - UML
 - 2.1 - Definition 4
 - Analysis Tool - Enterprise Architect 4
 - Build Systems 4
 - Design Systems 4
 - Design Tool - Enterprise Architect 4
 - Manage Complexity 4
 - Model Complexity 4
 - Model Structure, Manage - Enterprise Architect 4
 - Shared Model Development - Enterprise Architect 4
 - Support - Enterprise Architect 4
 - Visualize Systems - Enterprise Architect 4
 - UML Model
 - Add Connectors In Enterprise Architect, Tutorial 32
 - Add Diagram In Enterprise Architect, Tutorial 31

UML Model

- Add Element In Enterprise Architect, Tutorial 31
- Add Element To Diagram, Tutorial 31
- Add Package In Enterprise Architect, Tutorial 31
- Add View, Tutorial 30
- Define Connector Properties In Enterprise Architect 33
- Define Element Properties In Enterprise Architect 33
- Project Roles And Tasks In Enterprise Architect 38

UML Modeling Tool

- Enterprise Architect - Key Features 7

Upgrade

- Existing License (v. 6.5 And Earlier) 59
- Existing License (v. 7.0+) 59

User Forum 17

- V -

View

- Add To UML Model, Quick Start 30

Visio

- MDG Link For, Enterprise Architect 15

Vista

- Permissions For Enterprise Architect 25

Visual Studio 2005/2008

- MDG Integration For, Enterprise Architect 15

Visual Studio.NET

- MDG Link For, Enterprise Architect 15

Visualize Systems

- Enterprise Architect 4

- W -

Web Site

- Sparx Systems 16

What Is

- Enterprise Architect? 4

Windows Vista

- Permissions For Enterprise Architect 25

Work With

- Enterprise Architect 8

- Z -

Zachman Framework

- MDG Technology For, Enterprise Architect 15

Getting Started With Enterprise Architect

www.sparxsystems.com