**Enterprise Architect**

**手册**

**目录**

[Enterprise Architect 1](#_Toc318459820)

[手册 1](#_Toc318459821)

[1. 简介 4](#_Toc318459822)

[1.1. EA的特点 6](#_Toc318459823)

[1.2. 产品新增特性 6](#_Toc318459824)

[2. 操作介绍 8](#_Toc318459825)

[2.1. 一些基本的通用设置 8](#_Toc318459826)

[2.1.1. 设置默认代码环境 8](#_Toc318459827)

[2.1.2. 设置默认数据库 8](#_Toc318459828)

[2.2. 数据仓库 9](#_Toc318459829)

[2.2.1. 建立ODBC数据源 10](#_Toc318459830)

[2.2.2. 将EA示例导入工程 12](#_Toc318459831)

[2.2.3. 将工程导入数据仓库 13](#_Toc318459832)

[2.2.4. 导出工程 15](#_Toc318459833)

[2.3. 版本控制 17](#_Toc318459834)

[2.3.1. 在资源管理器中使用“TortoiseSVN”取得版本库数据 17](#_Toc318459835)

[2.3.2. 在EA中使用SVN 17](#_Toc318459836)

[2.4. 需求 20](#_Toc318459837)

[2.4.1. 需求属性 20](#_Toc318459838)

[2.4.2. 增加扩展属性 21](#_Toc318459839)

[2.4.3. 需求元素编号 22](#_Toc318459840)

[2.4.4. 需求元素列表视图 22](#_Toc318459841)

[2.4.5. 视图关系（可追溯关系） 23](#_Toc318459842)

[2.4.6. 变更需求 23](#_Toc318459843)

[2.4.7. 需求变更及需求缺限 24](#_Toc318459844)

[2.4.8. 使用维护元素 24](#_Toc318459845)

[2.4.9. 内部需求 25](#_Toc318459846)

[2.4.10. 需求图标颜色 26](#_Toc318459847)

[2.5. 用例图 26](#_Toc318459848)

[2.6. Class模型 27](#_Toc318459849)

[2.6.1. 创建“Class”模型 27](#_Toc318459850)

[2.6.2. 为模型中创建的“Class”对象使用别名 29](#_Toc318459851)

[2.7. 代码的生成 31](#_Toc318459852)

[2.7.1. 把整个包的设计类生成C＃格式代码 31](#_Toc318459853)

[2.7.2. 单个自动生成C＃代码 32](#_Toc318459854)

[2.8. 文档的生成 33](#_Toc318459855)

[2.8.1. Text文档的生成 33](#_Toc318459856)

[2.8.2. Html文档的生成 34](#_Toc318459857)

[2.9. 将包导入成XML形式 34](#_Toc318459858)

# 简介

**生命周期软件设计方案——Enterprise Architect是以目标为导向的软件系统。它覆盖了系统开发的整个周期，除了开发类模型之外，还包括事务进程分析，使用案例需求，动态模型，组件和布局，系统管理，非功能需求，用户界面设计，测试和维护等。**

****

|  |  |
| --- | --- |
| Business Process | 业务流程 |
| Requirements | 需求分析 |
| Use Case | 用例 |
| Domain Model | 领域模型 |
| Class | 类设计 |
| Database | 数据库设计 |
| Component | 组件 |
| Deployment | 部署 |
| Testing | 测试 |
| Maintenance | 维护 |
| Project Management | 项目管理 |
| User Interface | 用户接口 |

**注：红色部分是EA区别于powerdesign的功能**

利用EA，设计人员可以充分利用13 UML 2.0图表的功能——EA支持全部13种 UML 2.0图表和相关的图表元素；

包括：

结构图表：类、对象、合成元素、包、组件、布局；

行为图表：使用案例、通信、序列、交互概述、行为、状态、时序；

扩展：分析（简单行为）、定制（需求、变动和UI设计）。

EA提供使用工具，能够跟踪依赖关系、支持大型模型，帮助您管理大型复杂的工程；含有CVS或SCC提供工具，以时间快照为基线，通过比较来跟踪模型变动，从而实现版本控制；含有类似explorer的项目视窗，为您提供直观高性能的工作界面。

EA还含有一个所见即所得形式的模板编辑器，提供强大的文档生成和报告工具，能够生成复杂详细的报告，报告可以按照公司或客户要求的格式提供所需信息。

EA具备源代码的前向和反向工程能力，支持多种通用语言，包括C++, C#, Java, Delphi, VB.Net, Visual Basic和PHP，除此，还可以获取免费的CORBA和Python附加组件。EA提供一个内置的源代码编辑器，含语法突出功能，确保能够在一致的工作环境中快速导航和查找您的模型源代码。对于Eclipse或Visual Studio.Net工作人员，Sparx Systems还提供到这些IDE的轻量链接工具，您可以在EA中进行建模，而后直接跳转到自己偏爱的编辑器中进行源代码的进一步开发。代码生成模板还允许您对生成的源代码进行定制，使之同公司规范相符。

EA还提供对大多数软件开发语言和数据库架构的逆向工程支持，令您的应用程序可视化，从源代码、Java.jar文件甚至是.Net二进制汇编语言中获取完整框架。通过导入框架和库代码，实现对您已有投资重复利用的最大化。

EA还提供变换模板，编辑和开发均非常简单，支持先进的模型驱动结构体系（MDA）。通过内置的DDL, C#, Java, EJB 和 XSD变换，您可以从简单的“平台独立模型”开始来构建复杂的解决方案，并定位于“平台专门模型”（PSM）。一个PIM可以用来生成并同步多个PSM，使工作效率得到显著提高。

主要特性

UML设计和构建；

使用案例，逻辑，动态和物理模型；

定制进程模型的扩展或其他；

高品质的MS Word 兼容文档；

应用直观方便；

低许可费；

数据建模，前向DDL数据库工程，反向ODBC数据库工程；

多用户（Professional 和 Corporate Edition）；

前向或逆向代码工 程（Professional 和 Corporate Edition）——支持Java, C#, C++, VB.Net, Delphi, Visual Basic 和 PHP；

XML导入/导出容易；

拼写检查；

## EA的特点

1. 为整个团队提供高级的UML 2.0建模工具

EA为用户提供一个高性能、直观的工作界面，联合UML 2.0最新规范，为桌面电脑工作人员、开发和应用团队打造先进的软件建模方案。该产品不仅特性丰富，而且性价比极高，可以用来配备您的整个工作团队，包括分析人员、测试人员、项目经理、品质控制和部署人员等。

1. 丰富系统设计

Enterprise Architect是一个完全的UML分析和设计工具，它能完成从需求收集经步骤分析、模型设计到测试和维护的整个软件开发过程。它基于多用户 Windows平台的图形工具可以帮助您设计健全可维护的软件。除此，它还包含特性灵活的高品质文档输出。用户指南可以在线获取。

1. 快速、稳定、高性能

统一建模语言能够以一致方式构建强健和可跟踪的软件系统模型，而EA为该构建过程提供了一个易于使用和快速灵活的工作环境。

1. 端到端跟踪

Enterprise Architect提供了从需求分析、软件设计一直到执行和部署整个过程的全面可跟踪性。结合内置的任务和资源分配，项目管理人员和QA团队能够及时获取他们需要的信息，以便使项目按计划进行。

1. 在UML 2.0上构建

通过UML（统一建模语言），您可以构建严格的可追踪的软件系统模型。EA为UML构建软件模型提供了一个快速便捷的应用环境，它支持OMG定义的新UML2.0标准。

Enterprise Architect的基础构建于UML 2.0规范之上，不仅如此，使用UML Profile还可以扩展建模范围，与此同时，模型验证将确保其完整性。产品含有免费的extensions for BPMN 和 Eriksson-Penker profile，能够将业务程序、信息和工作流程联合到一个模型内。

## 产品新增特性

RTF所见即所得模板编辑器和报告生成器——EA 增强了其企业版和专业版中RTF文档生成功能，采用一个模板驱动方式生成RTF文档，允许用户在模型的元素层面上对其文档的外观、内容等进行定制。

1. 版本控制改进——对版本控件作了明显改进，支持对嵌套软件包的控制，这样，在导入基版本控制软件包时，就不会再覆盖嵌套软件包了。大型文件加载时间显著缩短，同时对AccuRev和Perforce的支持也得到改进。
2. 基线支持——企业版提供新的基线函数，在当前模型中，允许用户将一个完整的软件包分支保存为一个压缩的XMI文件。通过该基线函数，用户可以对当前软件包和存储的基线软件包进行比较。
3. MDA风格转换——EA专业版和企业版中提供MDA风格转换功能。MDA转换通过一个模板驱动方法将PIM（平台独立模型）元素转换为PSM（平台相关模型）元素。
4. 比较工具（Diff）——专业版和企业版提供该功能，允许用户对模型结构体系和保存的XMI进行对照比较。包括以下比较内容：基线函数创建的基线、保存在硬盘上的EA XMI文件、由EA XMI控制的版本。

导入二进制模块——EA专业版和企业版允许对Java Archive (.jar), . NET PE Files\* (.exe, .dll)和Intermediate Language (.il)二进制模块进行反向工程。

1. 构建和运行——EA Professional 和 Corporate版本支持该功能，为开发人员提供集成工具，将UML开发和建模同源代码开发和编译相集成。能够使用MDA变换从源类中生成nUnit和 jUnit测试类，并将测试过程直接集成到EA IDE中，如今还可以将UML和建模集成到构建/测试/运行/部署程序中。
2. MOF——在MOF 1.3和1.4标准下，允许向XMI导出包。向包中分配固定类型的元模型，可以创建MOF模型。MOF模型还可以输出为MOF 1.3或1.4 XMI文件规范。
3. 工程商讨论坛——确保整个团队始终处于最新状态。使用商讨论坛可以讨论您工程的开发和进度，同时为了讨论方便，还可以将工程内容进行分类，并制定标题。用户可以针对某个标题发出消息线程或响应消息线程。在论坛消息板中，您可以链接到图表的不同元素。在当前模型下还可以使用其他模型的商讨论坛。
4. Web服务——支持对W3C Web服务定义语言（WSDL）的前向和反向工程。支持对WSDL文件的建模、生成、输入操作。
5. XML架构（XSD）——支持对W3C XML架构（WSDL）的前向和反向工程。支持对XML架构的建模、生成、输入操作。
6. The Corporate Edition定位于大规模开发组。它支持Desktop和 Professional版本的所有功能，如连接到SQL服务器、MySQL, Oracle9i, PostgreSQL, MSDE，Adaptive Server Anywhere 和 MS Access backends以实现知识库共享。它还支持用户安全、用户注册、用户组、复制和元素的用户级同步。该版本还包含对MDG技术的支持，而且用户还也可以获得更多的资源。
7. The Professional Edition定位于工作组和开发人员。它支持通过复制和网络文件实现项目共享。专业版还含有ActiveX接口，可以查询EA项目、以XMI格式提取信息。它完全支持代码导入/导出，模型元素和源代码的同步以及对 SQL Server, MySQL, Oracle9i的逆向工程，而且包含对MDG技术的支持。
8. The Desktop Edition是为个体用户准备的，可以生成UML分析和设计模型。除代码工程（源代码和DLL导入/导出） 、Active-X接口和多用户模型共享之外，它可以实现专业版的所有功能。

# 操作介绍

## 一些基本的通用设置

### 设置默认代码环境

依次点击：“Tool->Options-Source Code Engineering->Default Language For Code Generation->JAVA”如图3所示：

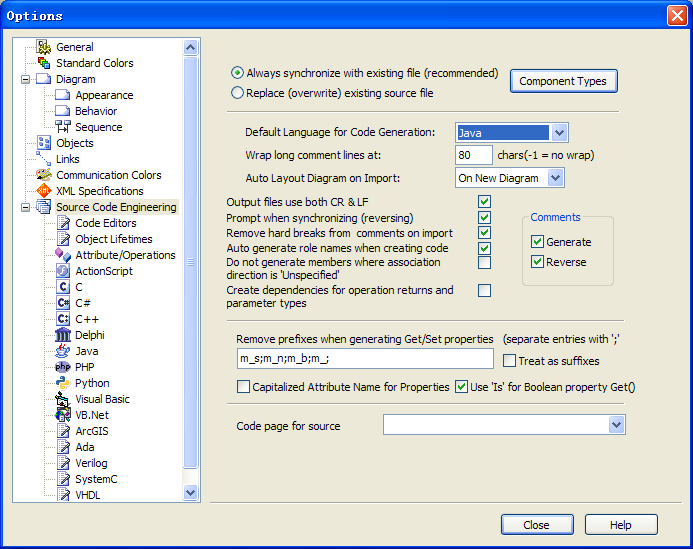


图-3

单击“Close”确认退出。

### 设置默认数据库

依次点击：“Tool->Options- Code Editors->Default Database->ORALE”

如图4所示：

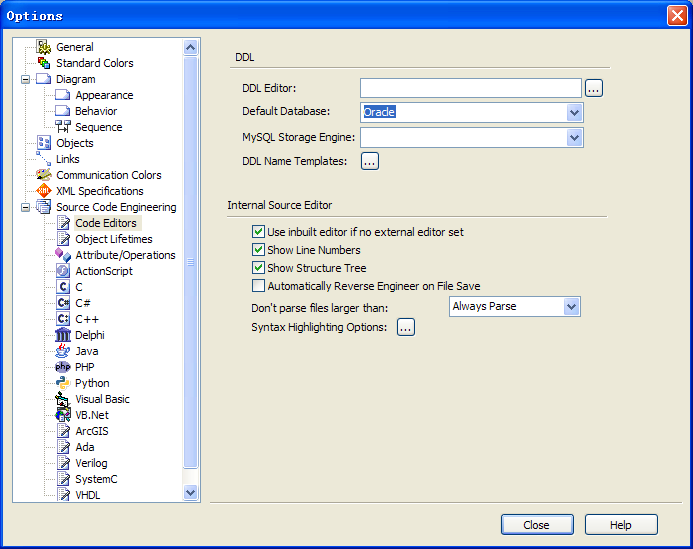
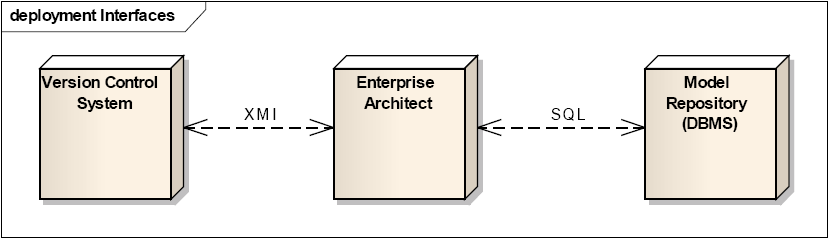


图-4

## 数据仓库

Figure 2 provides a high-level schematic of the relationship between Enterprise Architect and the external version control system. Note that versions are stored and retrieved as XMI files. XMI is used to serialize Package information to create a point-in-time snapshot. Enterprise Architect enforces that only a single user edits a given Package at once. This approach represents a “Lock-Modify-Unlock” solution, which helps to avoid revision conflicts – especially useful as XMI files are considered binary artifacts that cannot be merged directly via the Version Control System.

图二提供了一个EA与外部版本控制系统之间关系的高层次的示意图。注意：版本存取都是XMI文件。XMI被用于串行化包信息，以创建一个时间点的快照。EA强制同时仅一个用户修改特定的包。这类似于“加锁—修改—解除锁”解决方案（类似于VSS），借此避免版本冲突（这对XMI文件特别有用，考虑二进制部分它不能由版本控制系统直接合并）。



图二: EA与DBMS之间的数据交换

EA的仓储结构主要是将所有的UML元素集中控制于一个特定的RDBMS的数据库中，开发人员通过EA的客户端连到相同的一个仓库，第一个客户端所得到的，都是一个最终版本的开发模型。**(我的想法：估计只能保存最新的工程文件，每次可以生成最新的工程文件。但是真正的版本控制，还是由SVN这些工具来做)**

### 建立ODBC数据源

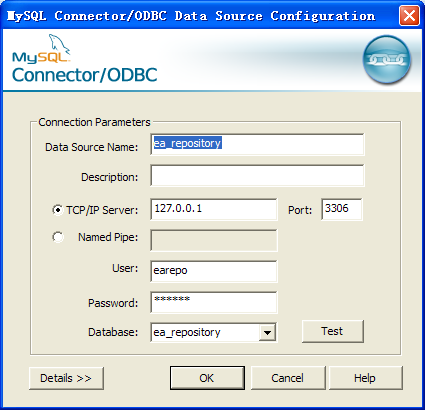
**必须先安装ODBC for MySQL驱动程序**

1. 打开 开始🡪管理工具🡪数据源（ODBC)，选择新增



1. 配置ODBC数据源





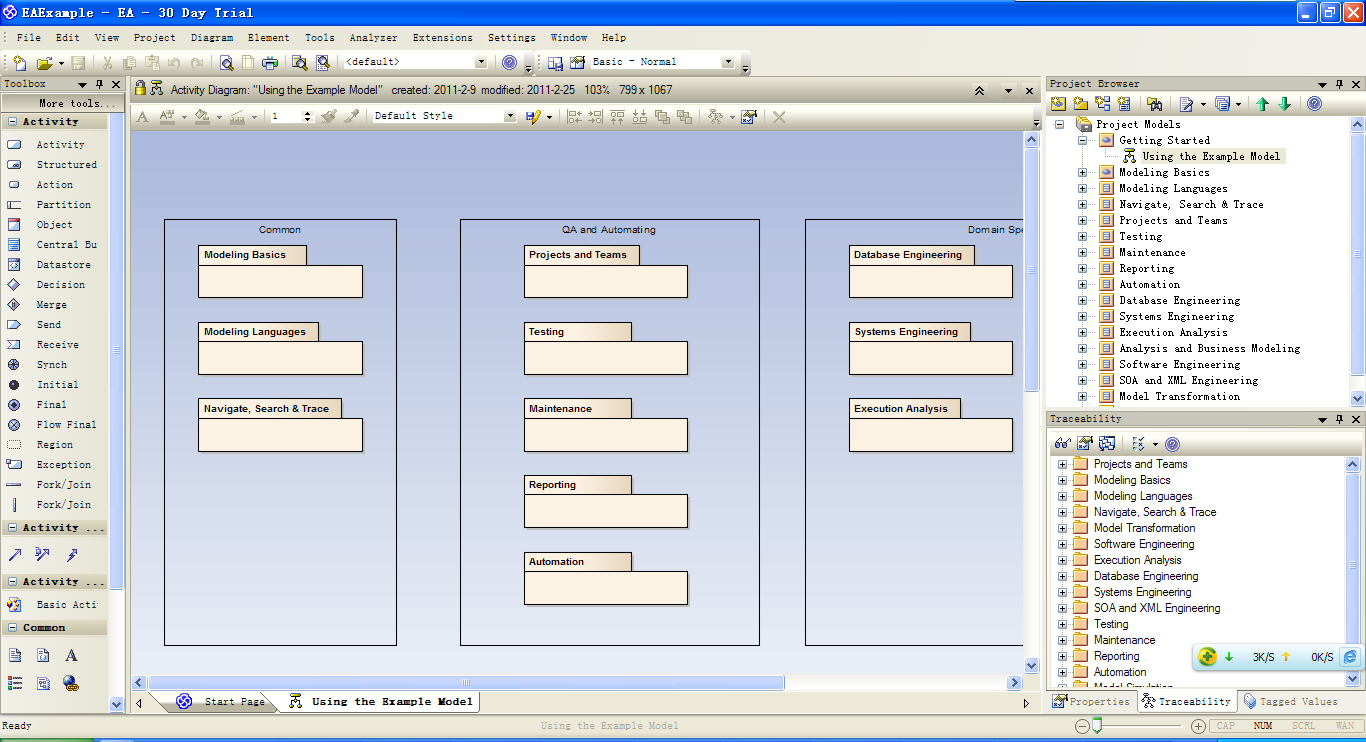
1. 测试连接

点击Test按钮，如测试成功将提示“Connection successful”



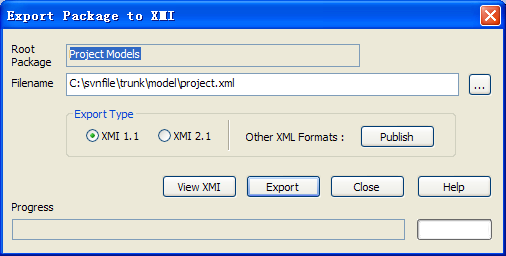
### 将EA示例导入工程

1. 打开EA示例工程



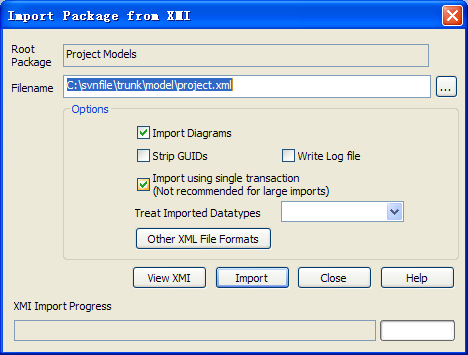
1. 导出EA示例数据

Project-->Model inport/export-->Export package to XMI



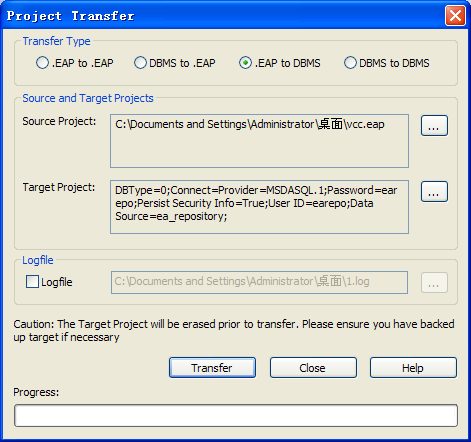
1. 打开VCC工程
2. 将数据导入EA示例数据

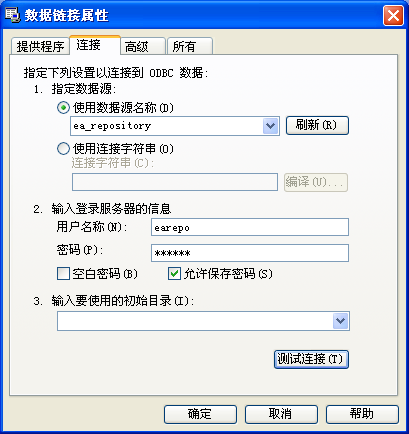
Project-->Model inport/export-->Inport package from XMI



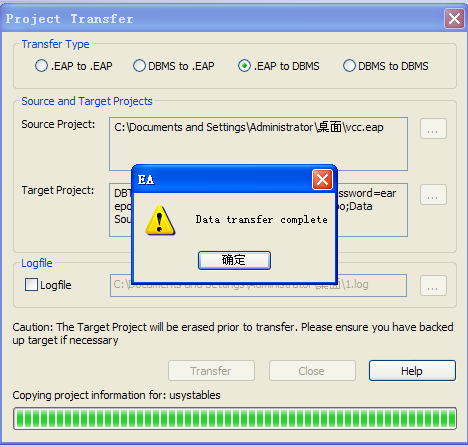
### 将工程导入数据仓库

1. Tools🡪Data Management🡪Project Transfer



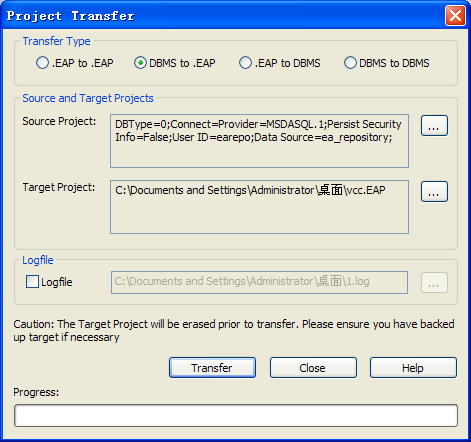


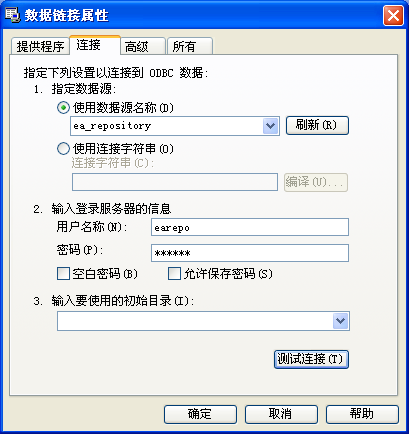
1. 导入工程



### 导出工程

1. Tools🡪Data Management🡪Project Transfer





## 版本控制

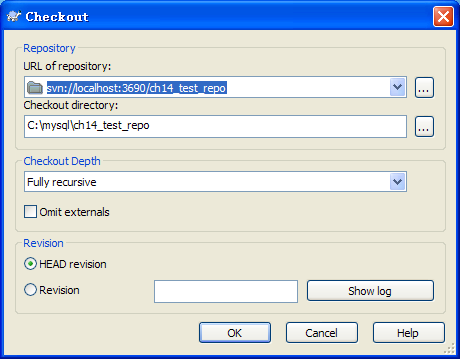
需要安装Subversion服务器端（只需要安装，不需要配制及启动服务）

Subversion客户端（只需要安装）

### 在资源管理器中使用“TortoiseSVN”取得版本库数据

在C:\svnfile目录下按鼠标右键，并选择"SVN Checkout"的选项，并在"Checkout"窗口的"文件库 URL"输入"svn://localhost:3690/ch14\_test\_repo"的路径：

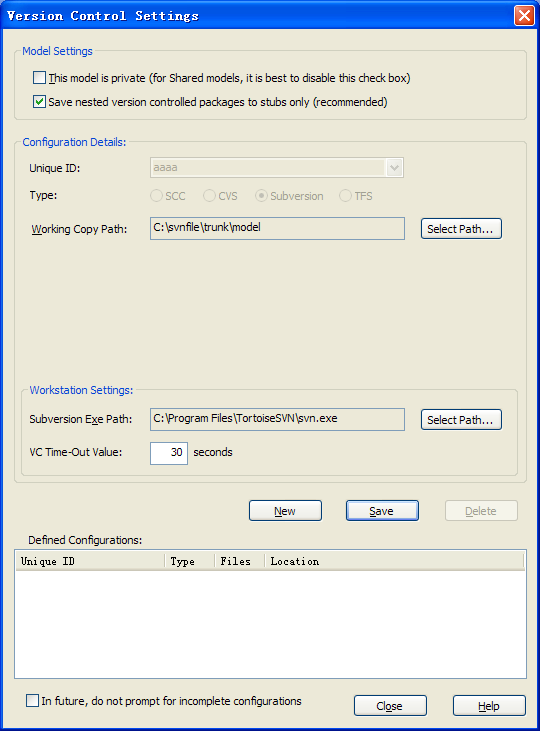
用户名、密码均为：admin/admin



### 在EA中使用SVN

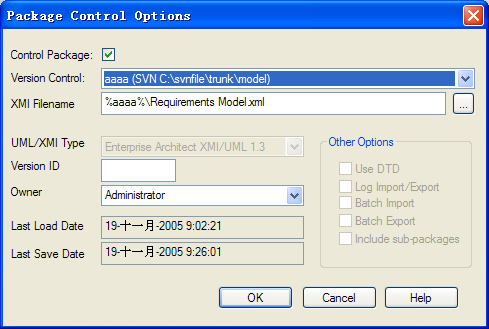
#### 在EA客户端设置Subversion相关信息

打开EA客户端（连接到MySQL对应的EA仓储），并选择"Project-->Version Control-->Version Control Setting"设置Subversion相关信息。

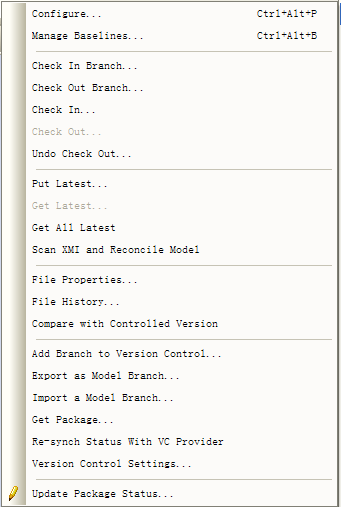


#### 将模型文件写入到Subversion版本库中

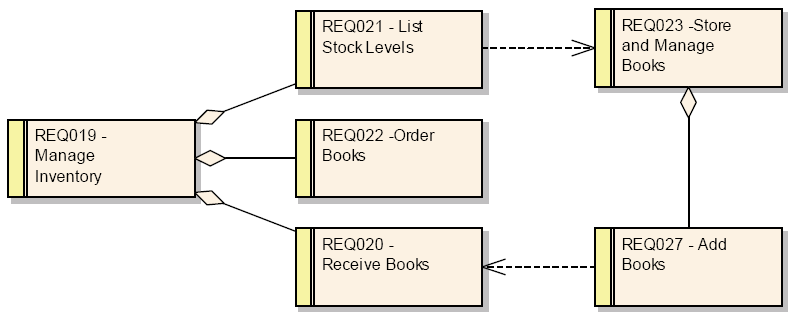
在做完整个项目对于版本控制的设置后，EA客户端就能以包(Package)为单位进行版本控制的设置



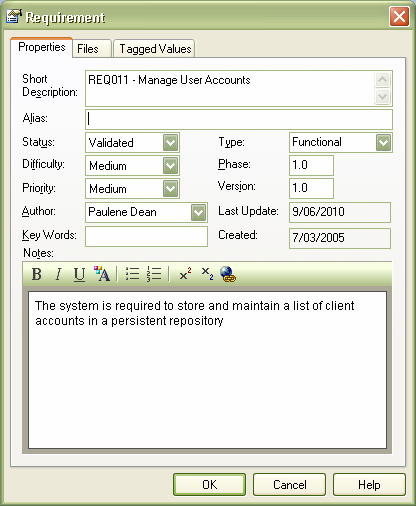
当完成上述步骤后，当选到已经设置完成版本控制的包后，其对应的"Package Control"目录将可以进行"Check In"或"Check Out"



## 需求



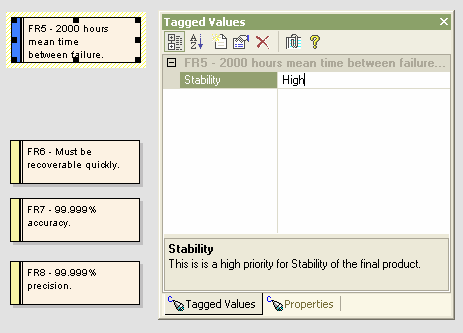
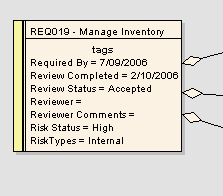
### 需求属性



* Every element, including a requirement element, that is part of a model has properties or attributes. *Enterprise Architect has built-in requirements attributes such as status, difficulty, priority, and type.*

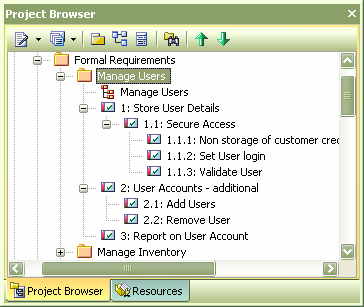
### 增加扩展属性

* It is common that there are a series of requirement attributes specific to any project. You can enter any number of additional attributes such as stability, cost, and lateness penalty through the use of tagged values.
* Tagged value data for an element is available as a separate window, which is accessed using Ctrl+Shift+6 (or from the main menu View | Tagged Values).

* To set this mode for a specific diagram, right-click on the diagram, and in the context menu, select: Properties | Elements | Show Compartments | [􀀹] Tags.

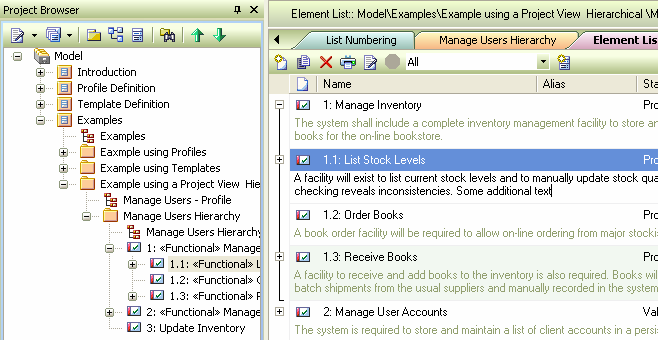
### 需求元素编号



Select a Package in the Project Browser

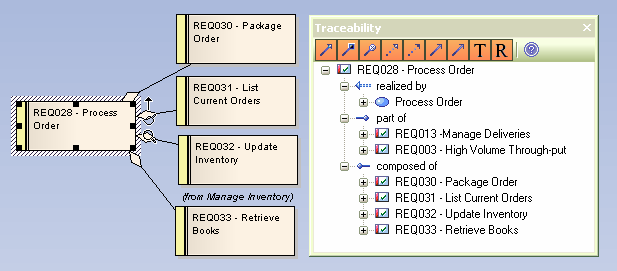
Right-click and from the context menu select: turn on Level Numbering

### 需求元素列表视图



* Select a Package in the Project Browser
* Right-click and from the context menu select: List View:

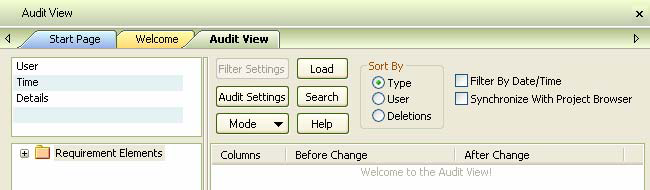
### 视图关系（可追溯关系）

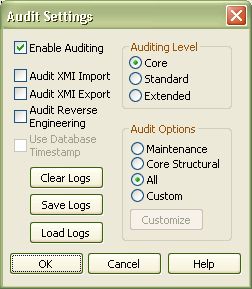


* Open the Traceability window (View | Traceability or Ctrl+Shift+4).
* Select an element to display its relationships.

### 变更需求

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





1. From the main menu select: Project | Audit View, which opens the following view:

2. Select the Audit Settings button.

3. This will bring up the Audit Settings window:

4. In the Audit Settings window set: [] Enable Auditing on as shown above.

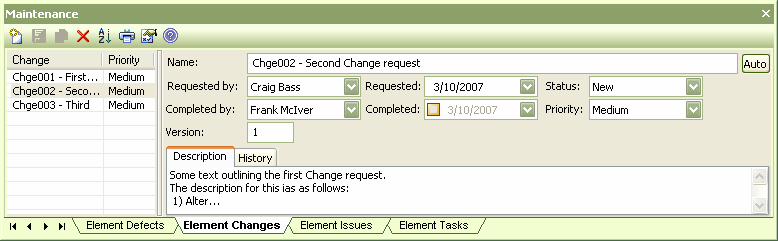
### 需求变更及需求缺限

Enterprise Architect supports logging of Change-requests against Requirements. This can be defined using two different methods:

a) Using the Maintenance View to list Changes, Defects, Issues and Tasks against each element.

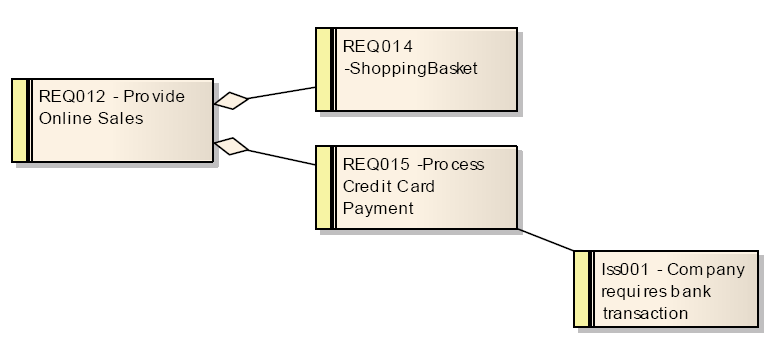
b) Using custom elements of type “Issue” and “Change” linked to the External Requirements being altered.

The Maintenance View can be accessed from the main menu using: View | Other Element Tools | Maintenance or (Alt+4). Figure 10 is an example of a set of changes listed for an element:



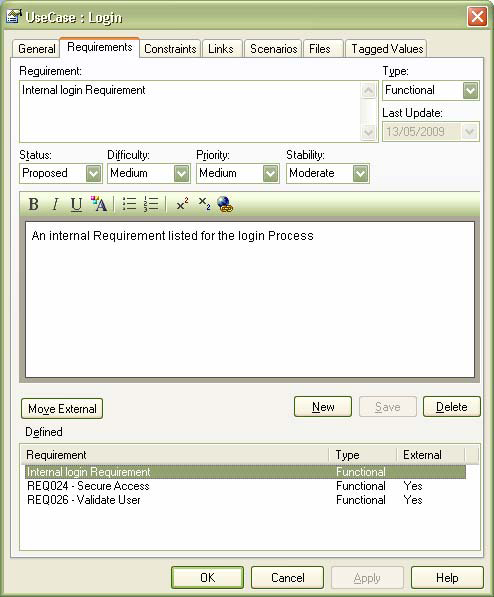
### 使用维护元素

Enterprise Architect’s maintenance elements include Elements of type: Issue and Change. These are accessible from: Toolbox | Maintenance or Toolbox | Custom.



### 内部需求

Enterprise Architect allows you to enter requirements at the level of an individual UML element. At this level these requirements can best be thought of as the ‘responsibility’ of the element.



Internal requirements can be externalized (see the [Move External] button above). This will create a new external requirement element with a realization relationship back to the original element (in this scenario – a use case element).

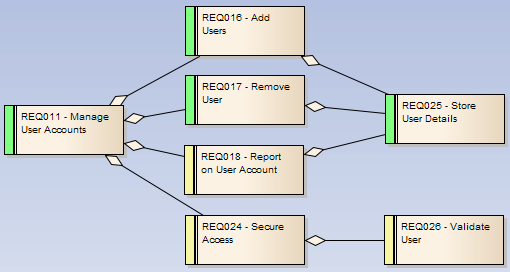
### 需求图标颜色

External requirements may be color coded to enable quick visual cues indicating the status of a requirement. To enable color coded external requirements take the following steps:

1. From the menu, select Tools | Options to open the local options window.

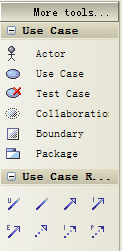
2. From the hierarchical tree select Objects.

3. Check the Show Status colors on diagrams checkbox.



## 用例图

Mode-->Use Case Model 下新建一张"Use Case Diagram"，命名为"信仁医院出院系统"





## Class模型

### 创建“Class”模型

依次点击“File->New Project->命名保存->在“Select model”中勾选“Class”（如图-5）

Enterprise Architect 7 入门教程转  - IT开拓者 - 飞.客

图-5

最后单击“确定”进入设计界面。

在设计界面的左边是工具箱（如图6）

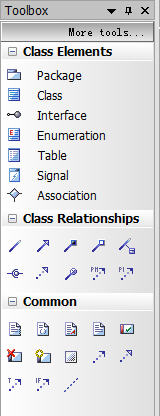


图-6

|  |  |  |
| --- | --- | --- |
| Class Elements | | |
| Package | 包 | |
| Class | 类 | |
| Interface | 接口 | |
| Enumeration | 枚举 | |
| Table | 表 | |
| Signal | 对象通信信号的描述 | |
| Association | 关联 | |
| Class Relationships | | |
| Associate | | 实现 |
| Generalize | | 泛化 |
| Compose | | 组成 |
| Associate Class | | 关联类 |
| Assembly | | 程序集 |
| Realize | | 实现 |
| Nesting | | 嵌套 |
| Package Merge | | 包的合并 |
| Package Import | | 包的导入 |
| Common | | |
| Note | | 注释 |
| Constraint | | 约束条件 |
| Document | | 文档：附加文档附件 |
| Artifact | | 人工：表示任意的系统任意的信息块 |
| Requirement | | 需求描述 |
| Issue | | 问题描述 |
| Change | | 变化描述 |
| Boundary | | 分界线 |
| Dependency | | 依赖关系 |
| Realize | | 实现 |
| Trace | | 追踪 |
| Information Flow | | 数据流 |
| Note Link | | 注释链接 |

表-2

### 为模型中创建的“Class”对象使用别名

首先在工具箱中点选Enterprise Architect 7 入门教程转  - IT开拓者 - 飞.客 ，然后在合适的位置左键单击确定，之后弹出属性窗口（如图7所示）

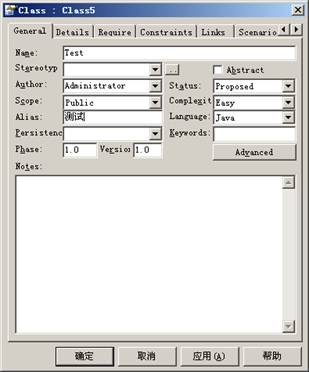


图-7

在“Name”栏中输入Class 的名称“Test”，然后在“Alias”中输入一个别名“测试”。其他相应设置好之后，确定退出属性窗口。

我们发现，在默认情况下“Class”所显示的名称是Test，但如果我们需要显示别名的时候，我们需要进行以下设置；

在类Test所在的包中找到class diagram，点击右键然后选择“Propreties…”进入如图8所示的窗口；



图-8

选择“Diagram”，进入如图9所示的窗口；



图-9

勾选“Use Alias if Available”，别名的显示即设置完成。



## 代码的生成

### 把整个包的设计类生成C＃格式代码

依次单击“Project->Source Code Engineering->Generate Package Source Code…（如图10所示）”

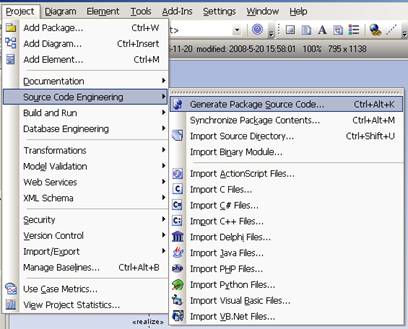


图-10

进入如图11所示界面：

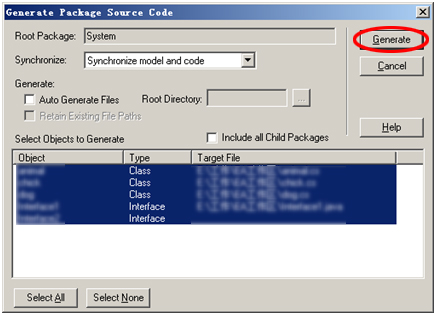


图-11

点击“Generate”，将包中所有的类成批转换为C＃文件。

### 单个自动生成C＃代码

右键单击需要转换为C＃文件的“Class”，在菜单中选择“Generate Code…”或者使用快捷键“F11”，弹出如图12所示菜单：

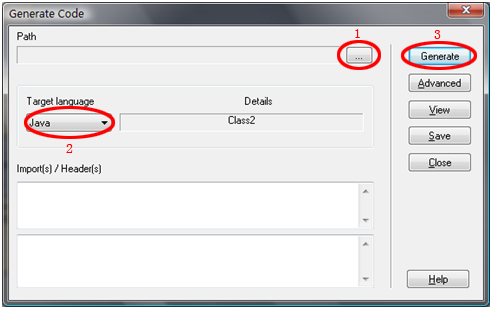


图-12

单击图中1的位置，选择将要生成文件的存储路径，然后在2的位置选择要生成什么语言的文件（在此我们点击下拉菜单选择C＃），最后点击3位置的“Generate”生成。

## 文档的生成

### Text文档的生成

依次单击“Project->Documentation->Rich Text Format(RTF) Report…（如图19）”或直接使用快捷键“F8”

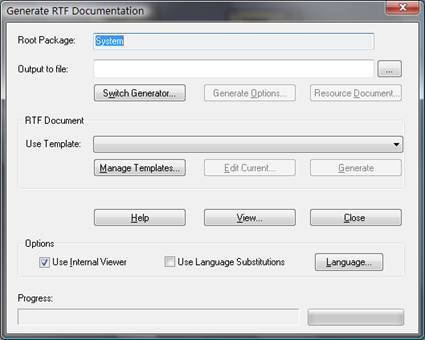


图-19

在“Output to file”后面单击，选择Text文档的生成路径，而后即可。

### Html文档的生成

依次单击“Project->Documentation->HTML Report…（如图16）”或直接使用快捷键“Shift+F8”，弹出图20所示窗口；

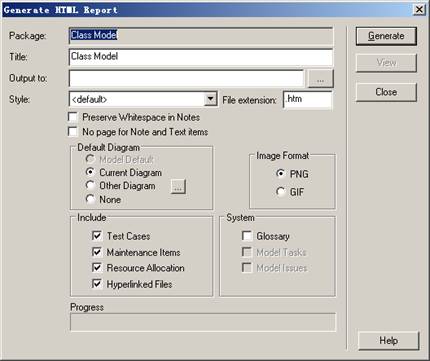


图-20

相关设置后，点击“Generate”生成即可。

## 将包导入成XML形式

依次单击“Project->XML Schema->Import XML Schema”如图21所示

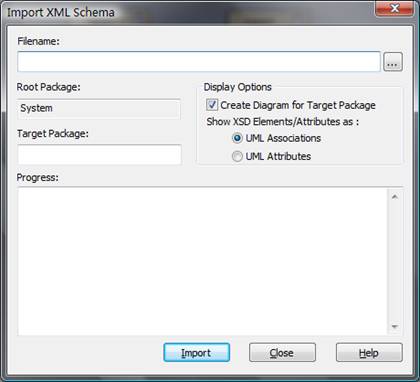


图-21

在“Filename”后面单击确定生成的XML文档的名称及路径，然后单击“Import”即可