

# Enterprise Architect 7.0入门教程

出自：<http://www.divsun.com/web/myblog/detail.asp?id=350>

## 一、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。

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。

## 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。

## 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。

## 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。

## 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。

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；

## 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，并可以方便地替换编辑和保存工程数据。专业版和企业版支持对模型元素进行自定义生成功能，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。

| 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。

| 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。

| 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。

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。

| 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。

| 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。

| 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。

| 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。

| 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。

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，而且用户还可以获得更多的资源。

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。

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。

二. 创建工程

依次点击“File->New Project->命名保存->在“Select model”中选择需要创建的类型（如图-1）->单击“OK”确认，得到界面如图-2所示

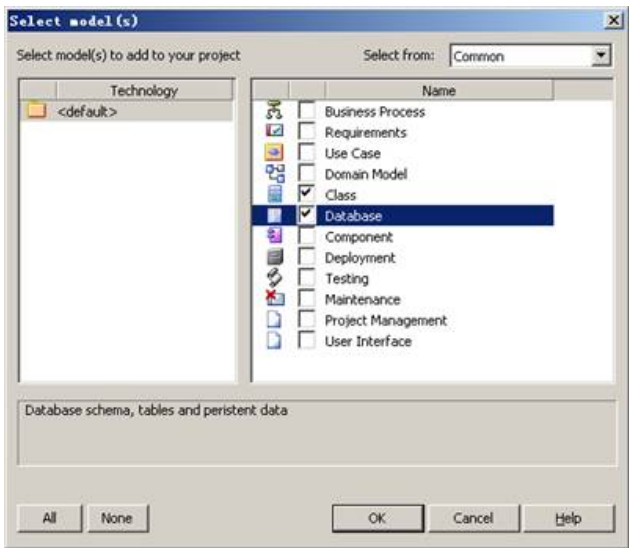


图-1

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

表-1

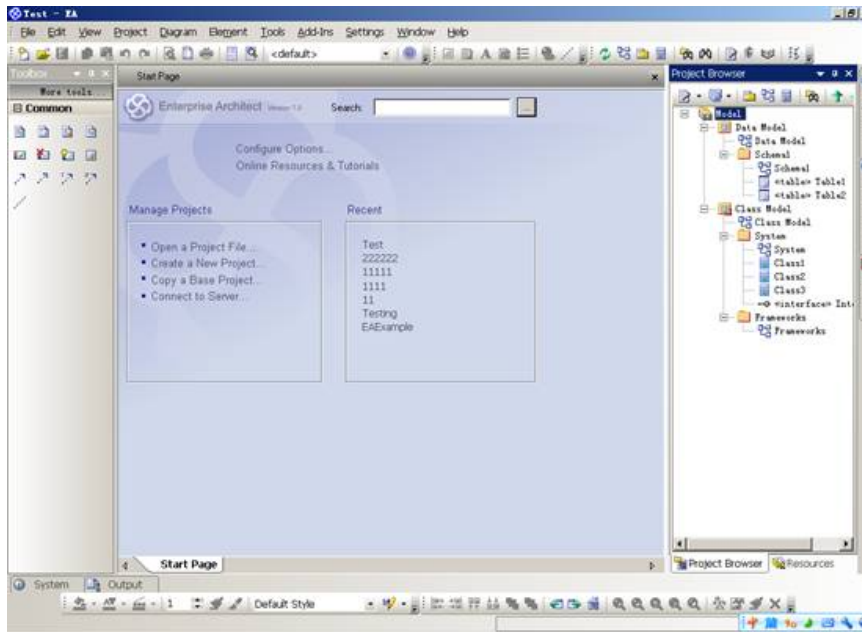


图-2

### 三. 一些基本的通用设置

#### 3.1 设置默认代码环境

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

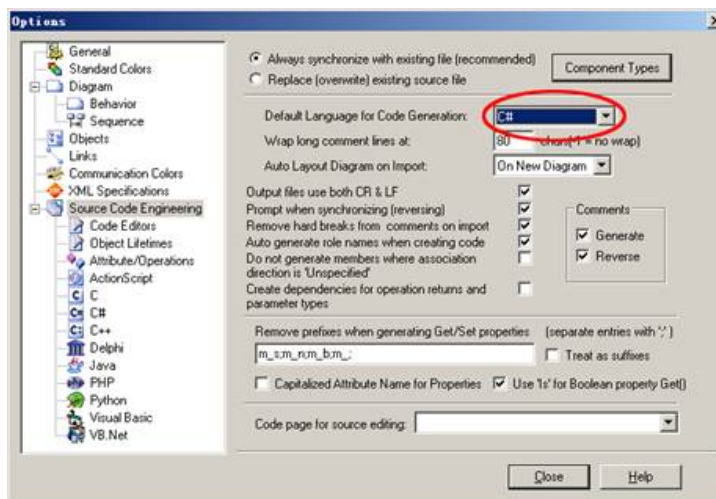


图-3

单击“Close”确认退出。

#### 3.2 设置默认数据库

依次点击：“Tool->Options- Code Editors->Default Database->SQL Server 2005” 如图4所示：

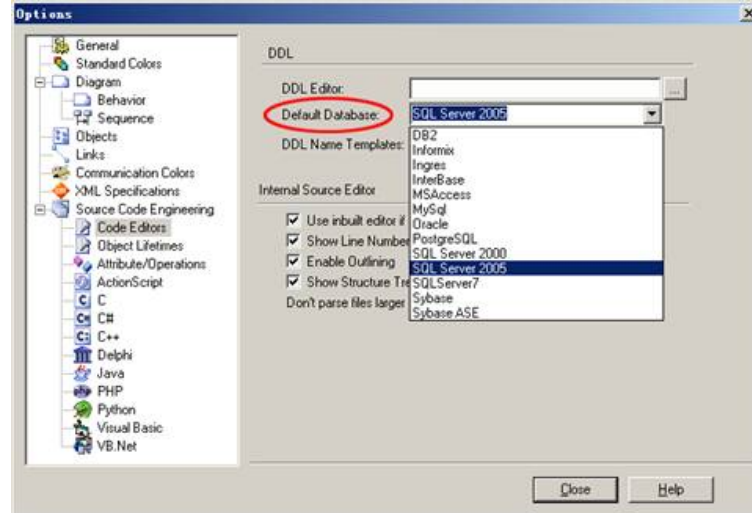


图-4

#### 四. Class 模型

##### 4.1. 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）



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）

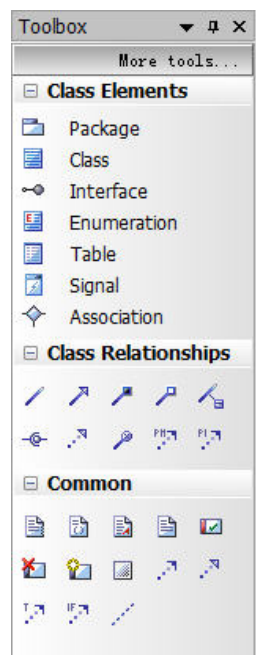




图-6

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
Nextion	嵌套

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

#### 4.2. 为模型中创建的“Class”对象使用别名

首先在工具箱中点选  Class，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）

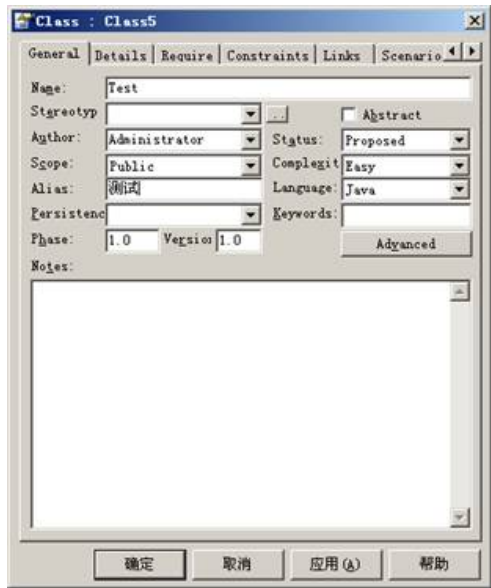


图-7

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；

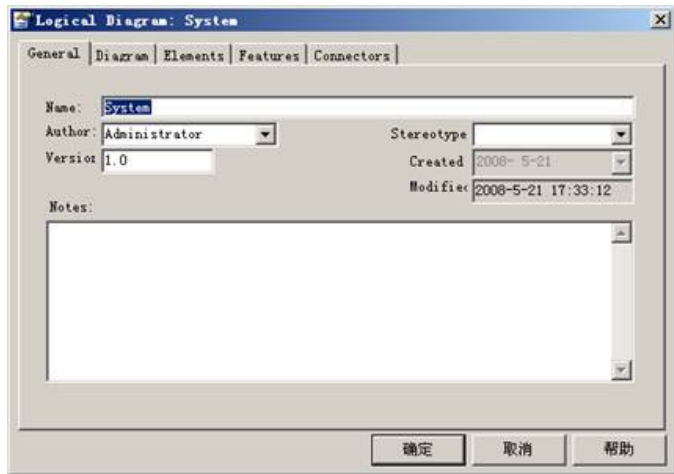


图-8

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。

## 五. 代码的生成

### 5.1. 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）”

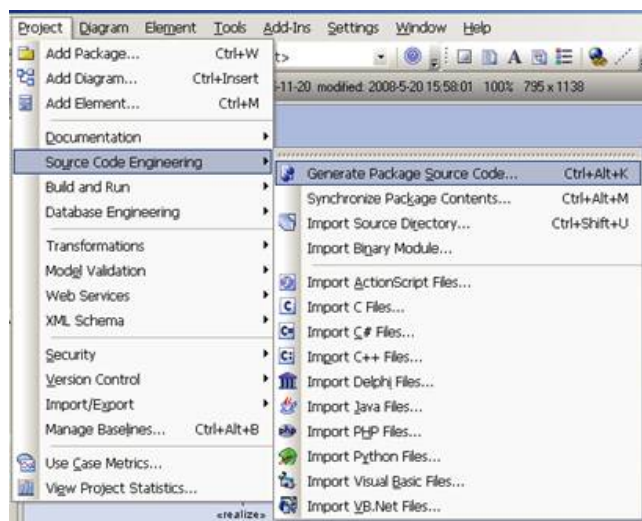


图-10  
进入如图11所示界面：

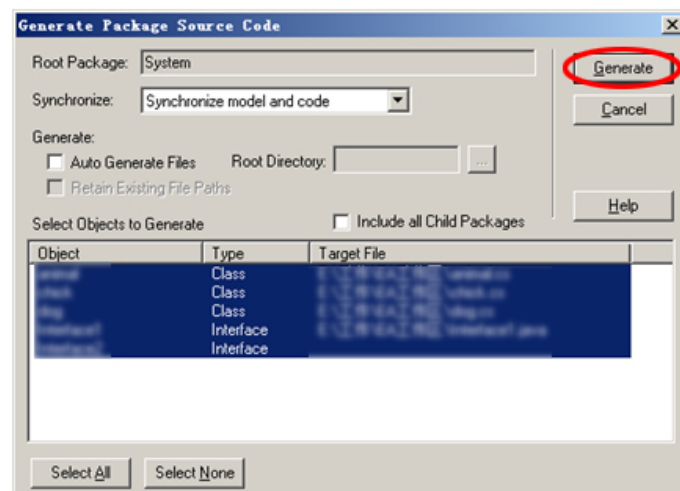


图-11  
点击 “Generate”，将包中所有的类成批转换为C#文件。

### 5.2. 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：



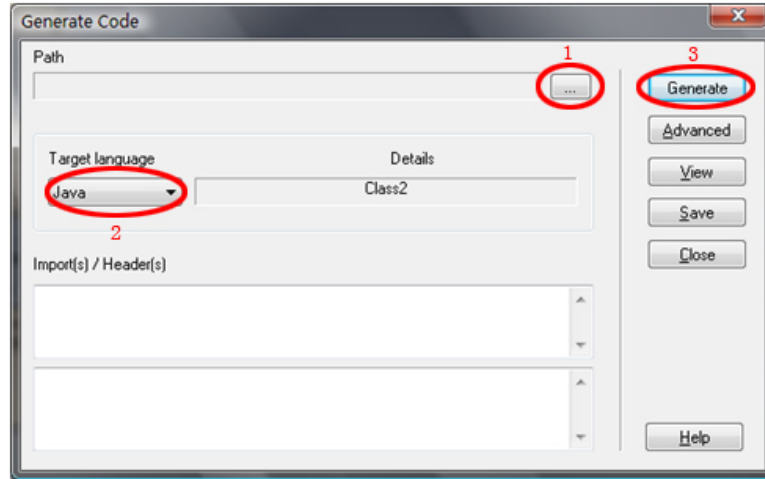


图-12

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。

#### 六. 设计模型与生成的代码同步

当生成了C#代码之后，接下来我们就将设计模型和生成的代码同步，步骤如下：

鼠标右键单击需要进行代码同步的对象，选择“Synchronize Model With Code...”或者使用快捷键“F7”，弹出如图13所示的对话框：

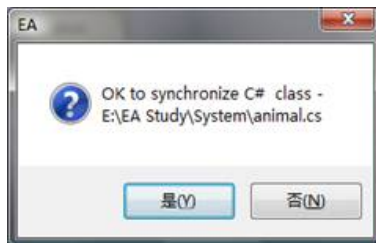


图-13

单击“是”，则我们就完成了生成的代码与设计模型进行了同步。

#### 七. 数据库模型，需求模型，组件，部署

##### 7.1. 建立数据表：

依次点击“File->New Project...->输入一个名称后保存->在右边选中Database（如图-14所示）”

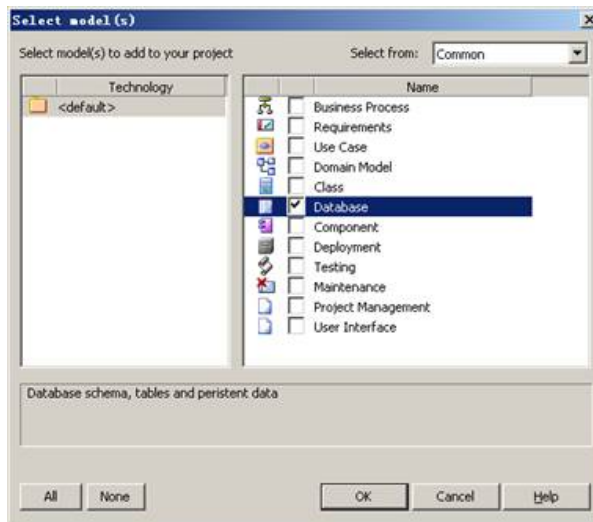


图-14

最后点击“OK”确认进入设计界面，“工程窗口”如图15所示：

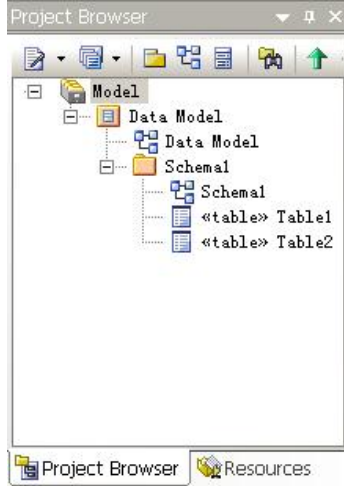



图-15  
工具栏及工具栏中的工具如图6及表2所示.

## 7.2. 建立数据表

在“工具箱”中单击  Table，然后将鼠标移入工作区域，此时鼠标变成  田状，单击鼠标，出现如图16所示的窗口.

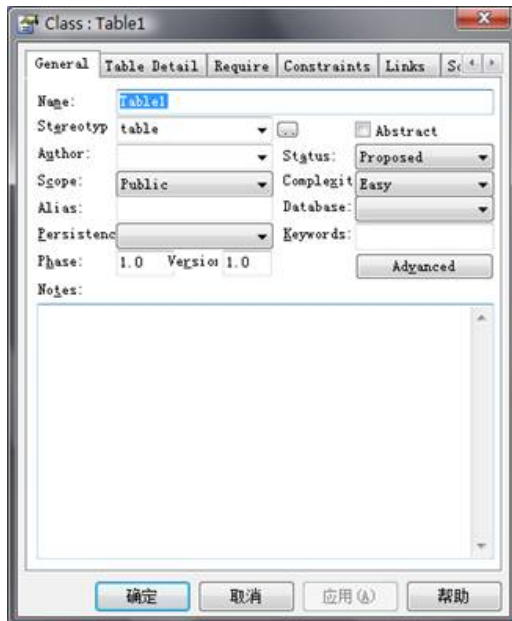


图-16

在进行一些表基本的设置（如表的命名“Name”等等）后，单击“Table Detail”进入表字段的设计（图17），



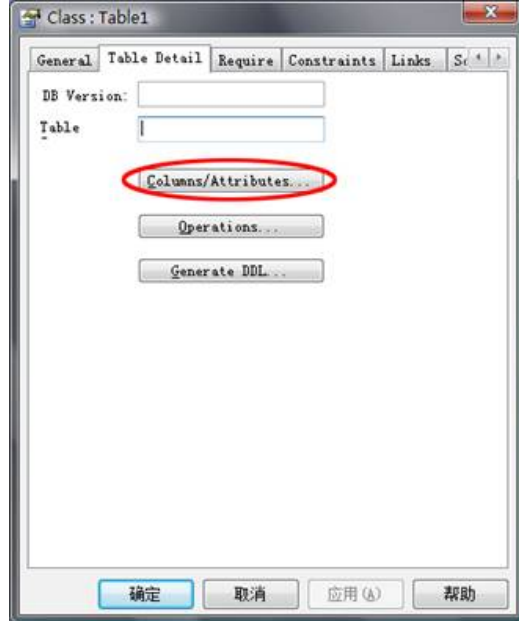


图-17

单击图14中红圈标识的“Columns/Attributes”进入表字段的设置。如图18所示。

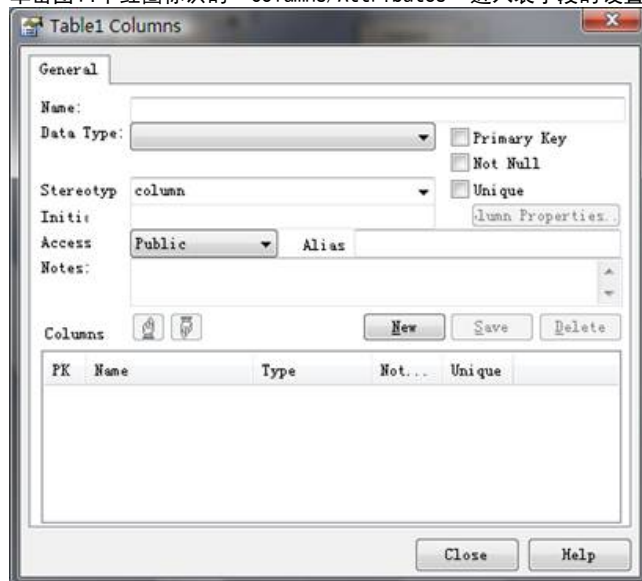


图-18

设置表字段名称“Name”，数据类型“Data Type”，及主键“Primary Key”，是否为空“Not Null”，等等，单击“Save”保存，然后单击“New”新建一个，就这样逐个设置表的字段。

## 八. 文档的生成

### 8.1. 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”

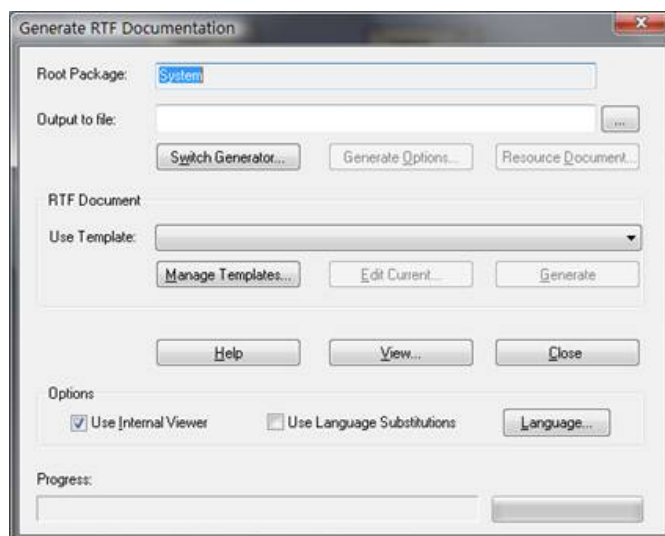


图-19  
在“Output to file”后面单击，选择Text文档的生成路径，而后即可。

## 8.2. 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；

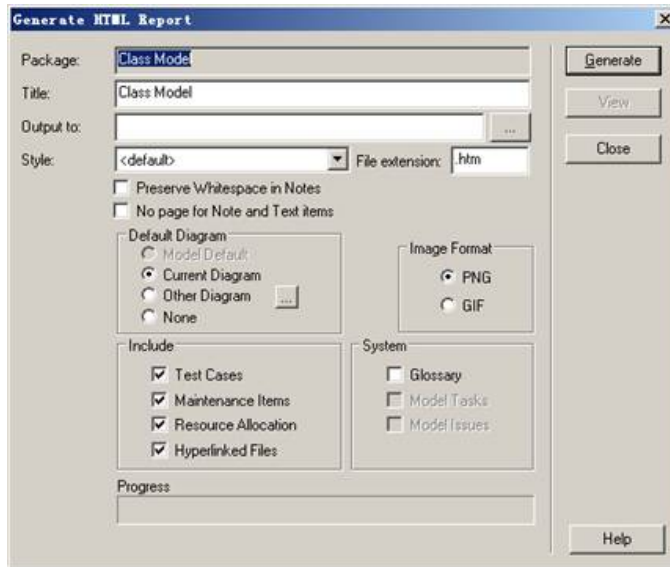


图-20  
相关设置后，点击“Generate”生成即可。

## 九. 将包导入成XML形式

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

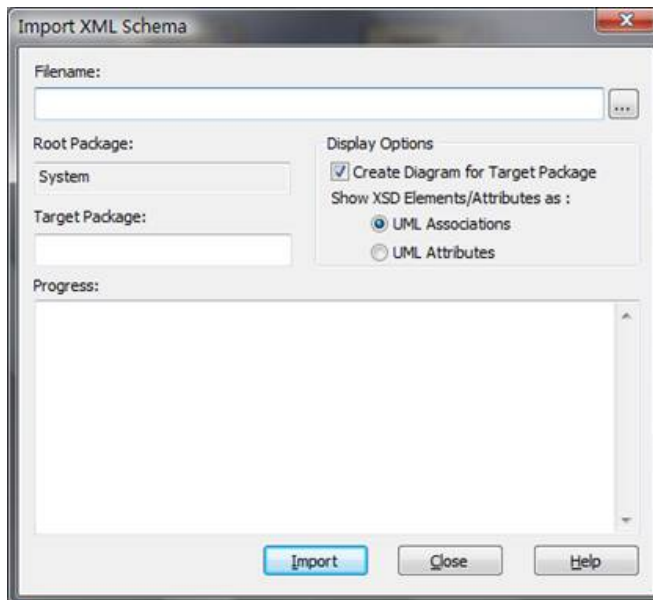


图-21  
在“Filename”后面单击确定生成的XML文档的名称及路径，然后单击“Import”即可

## 十. 画图的线的弯曲，自动布局，改变线的连接点的方法

首先在“活动图”中对“实线”点击右键（必须是实线，虚线不起作用），然后点击“Set Line Style”，而后“Bezier”，拖动中间的点调整线的弯曲。

Auto Routing(自动布局的方式)

Direct（改变线的连接点）

## 十一. 活动图及时序图的绘制

### 11.1. 活动图的绘制

在已建立的“Package”上右键单击->New----->New Diagram----->在弹出的窗口左边选择“UML Behavioral”（如图22所示）

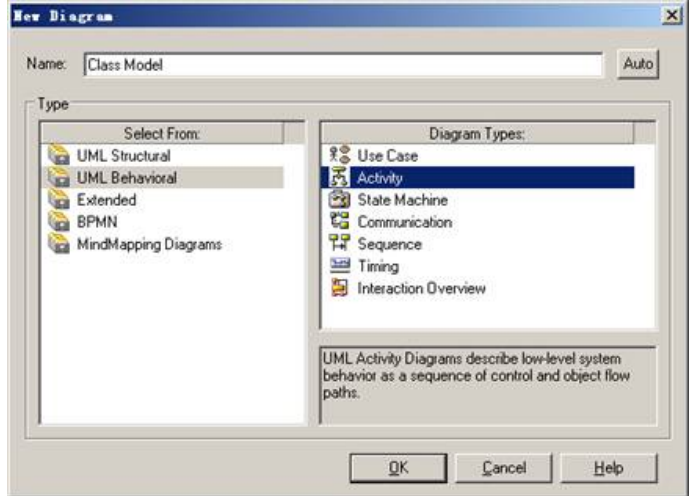


图22

在“New Diagram”的右边“Diagram Types”窗口中选择“Activity”，最后单击OK保存退出。此时，我们发现“Project Browser”窗口中增加了一个如图23所示的对象“Class Model”。

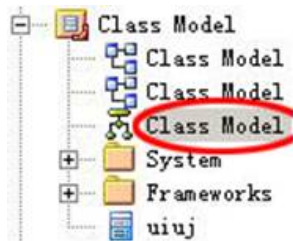


图-23

在左边的“Toolbox”中我们看到相关的工具。（如图24及表3所示）

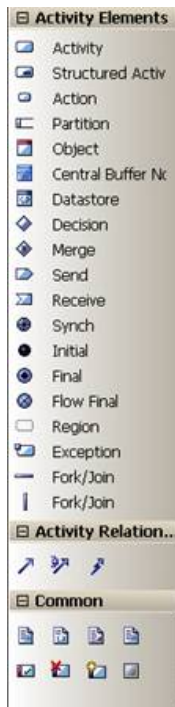


图-24

Activity	
Activity	活动状态
Structured Activity	嵌套活动
Action	动作
Partition	分区
Object	对象
Central Buffer Node	中央缓冲节点
Datastore	数据存储
Decision	判断
Merge	合并
Send	发送
Initial	最初的
Final	最终的

Flow Final	流结束
Region	区域
Exception	异常
Fork/Join	并发
Fork/Join	并发
Activity Relation...	
Control Flow	控制流
Object Flow	对象流
Interrupt Flow	中断流
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-3

## 11.2. 时序图的绘制

在已建立的“Package”上右键单击->New.....>New Diagram.....>在弹出的窗口左边选择“UML Behavioral”（如图25所示）

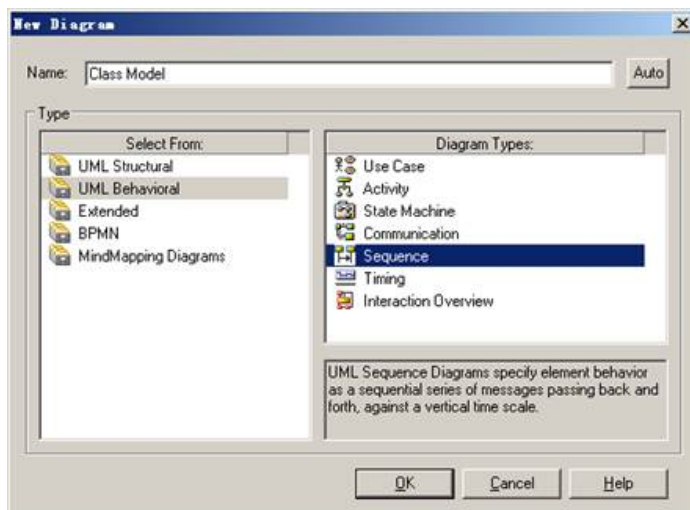


图-25

在“New Diagram”的右边“Diagram Types”窗口中选择“Sequence”，最后单击OK保存退出。此时，我们发现“Project Browser”窗口中增加了一个如图26所示的对象“Class Model”。

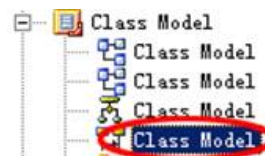


图-26

在左边的“Toolbox”中我们看到相关的工具。（如图27及表4所示）

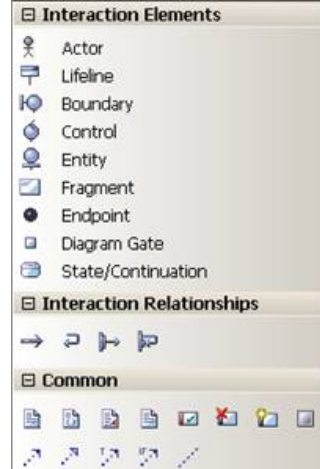


图-27

Interaction Elements	
Actor	角色
Lifeline	生命线
Boundary	
Control	控制
Entity	实体
Fragment	片段
Endpoint	端点
Diagram Gate	
State/Continuation	延时状态
Interaction Relationships	
Message	消息
Self-Message	安全消息
Call	呼叫
Recursion	递归
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-4

### 11.3. 如何添加注释


在Toolbox中点击  图标，然后在Workspace中左键单击你所需要添加注释的方位。弹出如图-28所示的窗口：



图-28

在空白区域填入注释内容，然后左键单击“OK”返回到如图-29所示的效果即可。

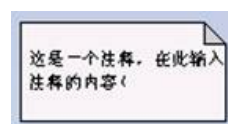


图-29

#### 11.4. 设置图中的字体样式

当我们在图中建立好元素后，使用下列方式对元素进行样式改变。图-30是在11.2的时序图的基础上增加了几个元素。

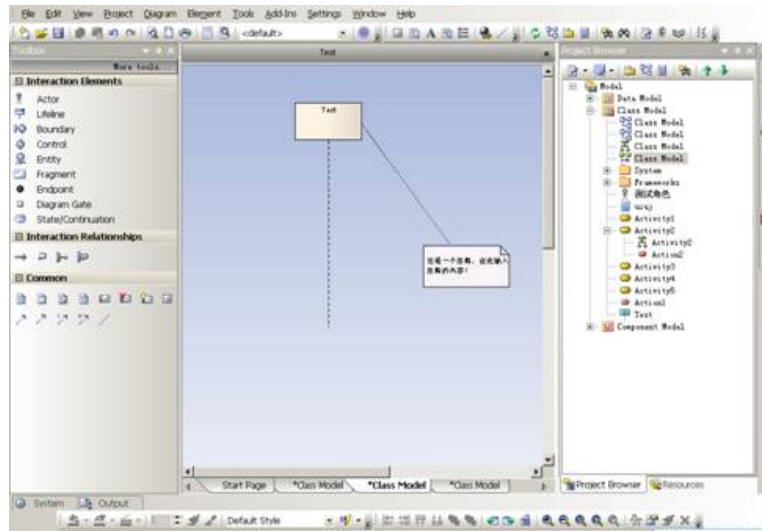


图-30

下面我们就来改变注释中的字体样式。

鼠标选中注释对象，下图是选中注释对象的前后对比：



图-31



图-32

#### 11.5. 将一个图的元素重新复制一份新的到其它图

首先新建一个图，然后在Project Browser中使用鼠标左键选中需要复制的元素不放鼠标左键拖到新建的图中即可。