**游戏工程配置**

当前版本:v 1.1

修改日期：2010-12-13

**修改记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Change Date** | **Change Reason** | **Change Content** | **Version** | **Create/Update** | **Approve** |
| 1 | 2010-10-20 | 新建 |  | 1.0 | 任东海 |  |
| 2 | 2010-12-13 | 修改 |  | 1.1 | 王定坤 |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |

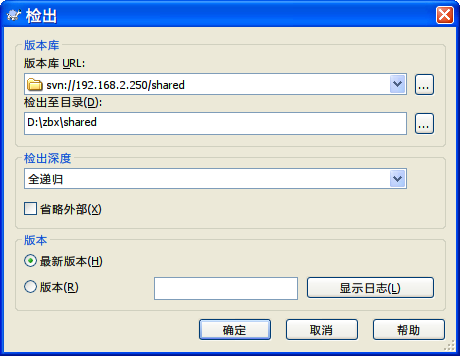
## 软件安装

配置游戏工程前，需要依次安装以下几个软件（安装文件[\\dc\程序\SVN](file:///\\dc\程序\SVN)下）：

1. **Tortoise SVN**（SVN客户端，用来管理日常的代码版本的管理）
2. **Tortoise SVN中文补丁包**(可选安装)
3. **AnkhSVN**(SVN的VS插件)

## 游戏工程基本配置

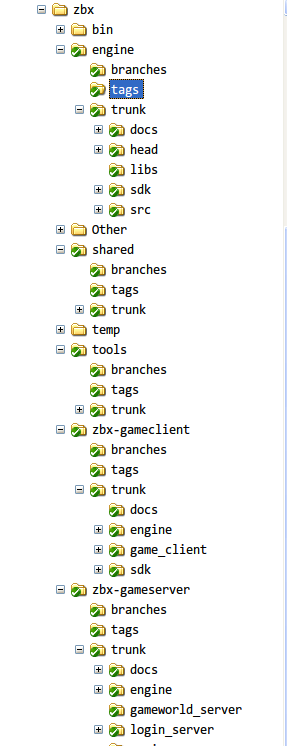
1. 首先为游戏项目新建一个根目录，以游戏项目的名字来命名，如project-x（下面均采用其来说明）
2. 在project-x目录下shared目录，用来存放客户端和服务器端公共的文件，以及一些编译时需要的全部头文件以及lib文件，包括dxsdk，maxsdk等等
3. 右击shared目录，选择“SVN导出”（英文版为SVN Export），弹出如下的对话框



在“版本库URL”框中填入：**svn://192.168.2.250/shared**

其它选项保持默认，点击确定后弹出用户认证界面，输入用户名和密码（为方便日后的操作，建议勾上“记住密码”选项），就可将代码从服务器签出到本地

1. 在project-x目录下面建立bin目录，这个目录用于存放服务器、客户端的资源文件和编译生成的服务器、客户端文件
2. 将shared\trunk\runtime目录下的所有文件复制到bin目录下，这些文件为服务器、客户端程序运行时所需的一些dll文件、配置文件等。
3. 库的命名方式：基于Multi-Threaded的运行库方式的库以\_mt后缀结尾，基于Multi-Threaded DLL的运行库方式的库以\_md后缀结尾，而每种方式均存在Debug和Release版本，为区分之，在Debug版本的后缀添加上d，这样共有四种版本：\*\_mtd.lib, \*\_mt.lib, \*\_mdd.lib, \*\_md.lib。
4. 整个项目所有工程，包含engine，tool，client，server四大部分，各个部分有着独立的权限，即不是所有开发人员都能访问所有工程，如客户端开发人员只能访问client，服务器开发人员只能访问server。
5. 整个项目所有工程（包含后面所讲的engine，tool，client，server等）编译出来的临时文件都存放在project-x\temp目录下,编译出来的最终文件存放在project-x\bin下面，这两个路径都不在svn的管理下，可以随便修改。您可以选择讲资源文件也存放到project-x\bin下面,然后直接在bin里面运行,或者可以在project-x下再建立一个data目录,将资源存放到data目录下,在vc里面指定工作路径到data目录来调试运行,也可以在bin目录下面创建一个exe的快捷方式,并将快捷方式的起始路径修改到data目录来直接运行。
6. 整个项目目录结构如下图。其中engine,game\_client,game\_server,tool这几个目录都是相互独立的，命名也可以随意.但是请保持目录的相对深度。



1. 如果您需要工作在别的分支版本下,请另外新建一组同样目录结构的目录,并从您工作的分支版本库下进行checkout操作。比如您的engine需要工作在一个叫alpha的分支下,您需要checkout如下目录结构

svn://192.168.2.250/engine/branches/alpha到project-x\_alpha/ engine

## 游戏客户端工程配置（客户端开发人员）

1. 在project-x目录下面建立gameclient目录，右击这个目录使用“SVN导出”，URL填入

**svn://192.168.2.250/zbx-GameClient**（注意大小写，因svn服务器是linux系统，文件名区分大小写），此处存放客户端的所有源代码以及其依赖库

1. 目录说明：

gameclient\trunk\docs目录下存放着客户端的一些文档

gameclient\trunk\engine目录下存放着客户端依赖的engine库的相关文件

gameclient\trunk\sdk目录下存放着客户端和engine依赖的第三方库的相关文件

gameclient\trunk\game\_client目录下存放着客户端的源文件

gameclient\trunk\game\_client.sln是客户端的项目文件

1. 游戏客户端工程由于历史原因，存在多种配置，但目前只需要Debug和Release版本配置，其它配置无效。代码生成采用Multi-Threaded的运行库方式，故Debug和Release版本均依赖于相应Multi-Threaded的运行库方式的engine库。

## 游戏服务端工程配置（服务器开发人员）

1. 在project-x目录下面建立gameserver目录，右击这个目录使用“SVN导出”，URL填入

**svn://192.168.2.250/zbx-GameServer**（注意大小写，因svn服务器是linux系统，文件名区分大小写），此处存放服务端的所有源代码以及其依赖库

1. 目录说明：

gameserver\trunk\docs目录下存放着服务端的一些文档

gameserver\trunk\engine目录下存放着服务端依赖的engine库的相关文件

gameserver\trunk\shared目录下存放着服务端和客户端共同依赖的文件

gameserver\trunk\gameworld\_server目录下存放着gameworld服务器的源文件

gameserver\trunk\ login\_server目录下存放着login服务器的源文件

gameserver\trunk\ region\_server目录下存放着region服务器的源文件

gameserver\trunk\SQL目录下存放数据库的数据表结构文件

gameserver\trunk\game\_server.sln是服务端的项目文件

1. 游戏服务端工程的代码生成采用Multi-Threaded的运行库方式，故Debug和Release版本均依赖于相应Multi-Threaded的运行库方式的net库（是engine库的一个组件）。

## 游戏工具工程配置（工具开发人员）

1. 在project-x目录下面建立tools目录，右击这个目录使用“SVN导出”，URL填入

**svn://192.168.2.250/tools**（注意大小写，因svn服务器是linux系统，文件名区分大小写），此处存放客户端的所有源代码以及其依赖库

1. 目录说明：

tools\trunk\docs目录下存放着工具相关的一些文档

tools\trunk\engine目录下存放着工具依赖的engine库的相关文件

tools\trunk\sdk目录下存放着工具和engine依赖的第三方库的相关文件

tools\trunk\tool\_character目录下存放着角色编辑器的源文件

tools\trunk\ tool\_max\_utility目录下存放着max导出插件的源文件

tools\trunk\ tool\_scene目录下存放着场景编辑器的源文件

tools\trunk\ tool\_ui目录下存放着UI编辑器的源文件

tools\trunk\tool.sln是工具的项目文件

还有一些其他的项目，在此不一一列举。

1. 游戏工具工程由于历史原因，存在多种配置，但目前只需要Debug和Release版本配置，其它配置无效。代码生成采用Multi-Threaded DLL的运行库方式，故Debug和Release版本均依赖于相应Multi-Threaded DLL的运行库方式的engine库。

## 游戏引擎工程配置（引擎开发人员）

1. 在project-x目录下面建立engine目录，右击这个目录使用“SVN导出”，URL填入

**svn://192.168.2.250/engine**（注意大小写，因svn服务器是linux系统，文件名区分大小写），此处存放客户端的所有源代码以及其依赖库

1. 目录说明：

engine\trunk\docs目录下存放着engine的一些文档

engine \trunk\head目录下存放着engine的头文件

engine \trunk\lib目录下存放着engine的库文件

engine \trunk\sdk目录下存放着engine依赖的第三方库的相关文件

engine \trunk\ src目录下存放着engine的源文件

engine \trunk\game\_client.sln是客户端的项目文件

1. Engine共分为8大模块：

Core是最基本的模块，为其他模块提供与操作系统相关的操作，以及一些最基本的公用函数；

Audio是音频模块；

FilePack是文件打包模块；

Graph是图形模块，封装了DX的一些操作；

Net是网络模块，封装着基本的Socket等操作；

UI模块提供大量与用户界面相关的控件；

Scene模块处理地形，场景物件，事件触发，环境因素等内容；

Character是角色模块，封装与角色相关的动画，材质，特效，骨骼，皮肤等等内容；

1. Engine项目需要生成四种版本的库文件，Multi-Threaded DLL Debug(\*\_mdd.lib)，Multi-Threaded DLL (\*\_md.lib)，Multi-Threaded Debug(\*\_mtd.lib)，Multi-Threaded (\*\_mt.lib)，其中前两种版本供Tool使用，后两种版本供客户端和服务器端使用。