**Envision Programming - Software Practices**

This document identifies best programming styles and practices for the Envision project.

Last Updated: June 23, 2010

Contents

[Programming Style/Practices 2](#_Toc265239302)

[Text Editor Setup 2](#_Toc265239303)

[Programming Conventions 2](#_Toc265239304)

[Directory Structure Conventions 3](#_Toc265239305)

[Source Control Practices and Conventions 4](#_Toc265239306)

[Installation: 4](#_Toc265239307)

[Initial Setup 4](#_Toc265239308)

[To get a copy of a directory in the repository: 4](#_Toc265239309)

[To Check in Files from Visual Studio. 5](#_Toc265239310)

[To Create a new Folder in SVN: 5](#_Toc265239311)

# Programming Style/Practices

## Text Editor Setup

**Tabs**: Set your Visual Studio editor to use tabs with a spacing or three characters for consistency with the existing Envision code base:

*Tools->Options->Text Editor->C/C++->Tabs->Smart*

*Tools->Options->Text Editor->C/C++->Tabs->Tab size: 3*

*Tools->Options->Text Editor->C/C++->Tabs->Indent size: 3*

*Tools->Options->Text Editor->C/C++->Tabs->Insert Spaces: Checked*

*Tools->Options->Text Editor->C/C++->Formatting->Indent Braces: True*

## Programming Conventions

**Blocks and Indentation:** All block should be enclosed by opening/closing braces at the same level as the block, e.g.

if ( doSomething )

{

a = b + c;

d = c \* c;

}

**Naming Conventions**:

|  |  |
| --- | --- |
| Convention | Example |
| Class names start in Uppercase and are mixed case | class MyClass; |
| Class Methods start in Uppercase and are mixed case | MyClass::MyMethod(); |
| Class members start with “m\_”, and are mixed case | MyClass::m\_myVarName; |
| Pointers start with “p” and are mixed case | float \*pFloatVar; |
| Class members that are pointers combine notation | float \*MyClass::m\_pFloatVar; |
| #defines are ALL UPPER CASE | #define DEFINE\_SOMETHING |

# Directory Structure Conventions

|  |  |
| --- | --- |
| Directory | Meaning |
| C:\Envision\ | root directory for all things Envision |
| C:\Envision\src\ | Root directory for source code – basically, everything in the Envision Solution file |
| C:\Envision\Docs\ | Root directory for Envision-related documents |
| C:\Envision\StudyAreas\ | Root directory for study areas binary and other non-edited files, e.g. IDU coverages. Each study area has its own appropriately named directory (e.g. c:\envision\StudyAreas\Lebanon) |
| C:\Envision\src\StudyAreas\ | Root directory for study area “editable” files that are included in the Envision solution, e.g. xml input files for plug-ins. |

The local directory structure mirrors the SVN structure exactly.

# Source Control Practices and Conventions

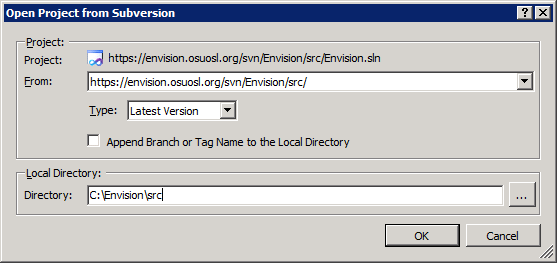
## Installation:

Install the AnkhSVN plugin for Visual Studio from <http://ankhsvn.open.collab.net/>

Install TortoiseSVN from: <http://tortoisesvn.tigris.org/>

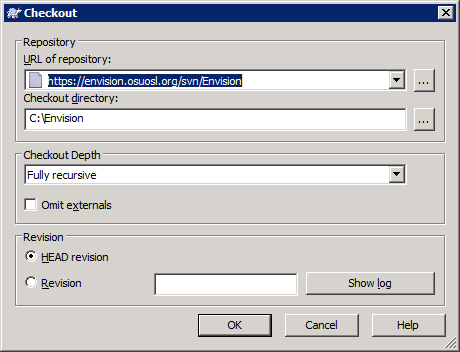
## Initial Setup

To get a new Visual Studio solution from SVN, bring up Visual Studio and select **File->Open->Subversion Project**. For an example getting the Envision project, see below.



## To get a copy of a directory in the repository:

1. Make sure the local folder exists on your machine
2. Right-click on the local folder and bring up the “Repo Browser” and navigate to the SVN folder you want to copy to your local machine
3. Right-click on the SVN folder and select **Check Out.**
4. In the Checkout dialog, specify the Checkout directory as the local folder and hit OK.



To get a copy of the entire Envision repository, create a local folder called c:\Envision and use the settings in the dialog box above.

## To Check in Files from Visual Studio:

Visual Studio automatically notes changes in checked out files. These can be seen on the “Pending Changes” panel in Visual Studio. To commit any changes back to SVN, select the desired files in the “Pending Changes” Panel and hit the “Commit” button.

## To Create a new Folder in SVN:

Bring up the Repository Browser, navigate to where you want to add the directory, right-click, and select “Create Folder”