Building VE-Suite for Windows (TAO Build)

This is our initial port of the new TAO based VE-Suite to windows. It must be built in Visual Studio 7.1(2003). Visual Studio 7.0 has many known issues and we can not provide support for it.

Dependencies

TortoiseSVN - Used to check out and update VE_Suite from ISU's code repository.

Visual Studio 7.1 (2003) - Used to compile the C++ code.

OpenGL Performer (3.1 or later) - Used to manage the scenegraph (the demo edition is fine).

VTK (4.4) - Used to render the visualization objects.

CMake - Needed to build the vtk project files.

VRJuggler (2.0alpha4.win32-vc71) - Used for management of the virtual environment.

VRJuggler dependencies zip file

Java 2 SDK - Needed for juggler's java-based VRJconfig program.

wxWidgets - Used to compile and run the gui. (Click "download" on the left side menu to get to the files)

Xerces - Used to read and write XML. (download xerces-c-src_2_5_0.zip)

ACE+TAO - Used for cross platform communications.

Notes on installation of dependencies

- After TortoiseSVN is installed, you can check out the code provided you have an account registered with ISU's repository. Create a VE_Suite folder. Go to VE_Suite folder and right-mouse click, choose "Checkout...". For URL, enter

  https://subversion.vrac.iastate.edu/svn/TSVEG/VE_Suite

- Alternatively, if you get the VE-Suite source code as a compressed tarball, some unzip utilities may not maintain the correct directory structure. This can be fixed by unzipping the files on a Unix-based system and then moving the files to the windows machine.

- While the default locations for installations will work, we recommend creating a directory in the root (C:\) directory to minimize the path lengths and the accompanying potential for typos when configuring environment settings.

vtk-specific notes

- After downloading the compressed vtk source code and the CMake installer, unzip the vtk source to C:\<your_path>\VTK-4.4-LatestRelease and run the CMake setup program (e.g., CMSetup205.exe) to install CMakeSetup.

- Go to C:\<your_path>\VTK-4.4-LatestRelease. You will have a "VTK" folder containing the vtk source code. Create two additional folders: "build" and "install"

- Run CMakeSetup, and enter C:\<your_path>\VTK-4.4-LatestRelease\VTK as source code location and C:\<your_path>\VTK-4.4-LatestRelease\build as the place to build the binaries. Check the box "Show Advanced Values" and press "Configure"

- Change some options from their default values:

    1. BUILD_SHARED_LIBS to ON

    2. BUILD_TESTING to OFF

    3. CMAKE_INSTALL_PREFIX to C:\<your_path>\VTK-4.4-LatestRelease\install

    4. VTK_USE_HYBRID to ON

    5. VTK_USE_PARALLEL to ON

- Press "Configure" (possibly need to do it twice), then "OK"

- Go to C:\<your_path>\VTK-4.4-LatestRelease\build. Double click on "VTK.sln"

- Left-click on ALL_BUILD, then right-click and select "Build"

- Required "hack": When the build is finished, go to C:\<your_path>\VTK-4.4-LatestRelease\build\CMake\Debug and move the .dll files up one level.

- Return to Visual Studio and left-click on INSTALL, then right-click and select "Build"

- For wxWidgets, the .dsw file is in the src directory, and will need to be converted to a .sln file using Visual Studio 7.1

- For Xerces, the .sln file is in Projects\win32\VS71. You may be asked to convert file formats, select yes to all.

When building ACE and TAO

- You need to create a file called "config.h" in the ACE_wrappers\ace directory, with the line '#include "ace/config-win32.h"' in the file.

- ACE_ROOT and TAO_ROOT must be established and ACE_ROOT\bin placed in your $PATH before building ACE and TAO. This can be done using a batch file (.BAT) with these lines:

```
set ACE_ROOT=C:\ACE_wrappers

set TAO_ROOT=%ACE_ROOT%\TAO

set Path=%Path%;%ACE_ROOT%\bin

"C:\Program Files\Microsoft Visual Studio .NET 2003
\Common7\IDE\devenv.exe" ..\ace.sln
```

- The last line is modified for each solution file: ACE.sln, TAO.sln and TAO_idl.sln.

Environment Settings

To specify the build/run environment, perform a one-time edit of the environment variables that are defined in the setup batch file ($(VE_SUITE_HOME)\VE_Installer\setup.bat). If the paths of your environment variables contain spaces, you need to put the directory containing the spaces in quotation marks.

Example: `set VTK_HOME=C:\"Documents and Settings"\user\Desktop\vtk`

Building VE-Suite

To set the defined environment variables and launch a session of Visual Studio with the VE-Suite project loaded, run the batch file:

```
$(VE_SUITE_HOME)\VE_Installer\build.bat
```

Currently there are four supported configurations available: Debug_TAO, Release_TAO, Debug_OMNI and Release_OMNI.

Running VE_Xplorer

Edit the line in your $(VE_SUITE_HOME)\VE_Installer\omniOrb4.cfg that contains "InitRef" to be:

```
InitRef = NameService=corbaname::localhost:2809
```

You need to have omniNames started to run VE_Xplorer. If omniNames is not running, double click on $(VE_SUITE_HOME)\VE_Installer\omninamesStart.bat, or type in a command window,

```
omniNames -start 2809
```

To run the solution from within Visual Studio, change the working directory in the Configuration Properties tab to point to the directory where your VE-Suite parameter file is located (for example, change the working directory to $(VE_SUITE_HOME)\VE_TestSuite)

You will also need to pass in your juggler config files as command arguments on the configuration properties tab. With the two projects still open, right-click on the VE_Xplorer project file, and select properties. For simulation mode, you add the following to your command arguments section:

```
$(VE_SUITE_HOME)\VE_Xplorer\vjconfig\sim.base.jconf
$(VJ_BASE_DIR)\share\vrjuggler\data\configFiles\sim.wand.mixin.jconf
```

After you enter a parameter file name and the scene loads, double click on $(VE_SUITE_HOME)\bin\ve_conductor.bat to bring up the gui.

Modified 3/4/2005