Instructions for Compiling GiPSiNet (v.2.0.00_28102008) using Visual Studio .NET 2005

1. Intel Math Kernel Libraries v8.0

Note: the directory and path will be different, if using different version of MKL.

a. Copy the Intel Math Kernel Libraries License File to “C:\Program Files\Common Files\Intel\License\”.
b. Install the Intel Math Kernel Libraries to default directory, usually is “C:\Program Files\Intel”, with registering PATH, LIB, and INCLUDE Environment Variables option selected.
c. You may also use “C:\Program Files\Intel\MKL\8.0\tools\environment\mlkvars32.bat” for environment variable settings.

2. POSIX Threads v2.8.0

Note: currently, the pthread supports only 32 bit

a. Download POSIX Threads library from: http://sources.redhat.com/pthreads-win32/
b. Copy “pthreadVC2.lib” into the directory “C:\Program Files\Microsoft Visual Studio 8\VC\lib”
c. Copy “pthreadVC2.dll” into the directory “C:\Windows\System32\”
d. Copy “pthread.h”, “sched.h”, “semaphore.h” into the directory “C:\Program Files\Microsoft Visual Studio 8\VC\include”
e. For compile GiPSiNet, the source code may needs some modification due to definition conflict with ACE/TAO as the following:

```
pthread.h, comment the following lines:
        /*
        struct timespec {
          long tv_sec;
          long tv_nsec;
        };
        */
```

```
semaphore.h, change the following lines:
        #ifndef HAVE_MODE_T
                typedef int mode_t;
        #endif
```

3. Xerces-C++ XML Parser v3.0.0

a. Download Xerces-C++ XML Parser library from: http://xml.apache.org/xerces-c
b. Copy “xerces-c_static_3.lib” and “xerces-c_static_3D.lib” into the directory “C:\Program Files \ Microsoft Visual Studio 8\VC\lib”
c. Copy “xerces-c_3_0.dll” and “xerces-c_3_0D.dll” into the directory “C:\Windows\System32\”

Updated: October 16, 2008

d. Copy the contents of the “include” directory into the directory “C:\Program Files\ Microsoft Visual Studio 8\VC\include”
e. Define “XERCES_STATIC_LIBRARY" in preprocessor macro
f. Enable "Treat wchar_t as a builtin" on xerces-all project and rebuild

4. GLUT v3.7

a. Download GLUT libraries from: http://www.xmission.com/~nate/glut.html
b. Copy “glut.h” into the directory “C:\Program Files\Microsoft Visual Studio 8\VC\include\GL”
c. Copy “glut32.lib” into the directory “C:\Program Files\Microsoft Visual Studio 8\VC\lib”
d. Copy “glut32.dll” into the directory “C:\Windows\System32\”
e. For some reason, if the GiPSiVisualization can not compile, edit “glut.h” as the following: add `“__declspec(noreturn)”` between `“_CRTIMP” “void __cdecl”.` The result will be `“__extern _CRTIMP __declspec(noreturn) void __cdecl exit(int);”`

5. OpenHaptics Toolkit

a. Download OpenHaptics toolkit from: http://dsc.sensable.com, register is required for academic edition.
b. Install PHANTOM Omni device driver
c. Install OpenHaptics toolkit
d. Recompile “hdu.lib” and “hdud.lib” which are release and debug version from “C:\Program Files\SensAble\3DTouch\utilities\src\HDU” directory by setting Runtime Library to “Multi-threaded”
e. For compiling GiPSiNet, set Preprocessor Setting to
   In Application, Add “GIPSI_HAPTICS_ENABLED”
   In Toolkit, Add “_USE_PHANTOM” and “GIPSI_HAPTICS_ENABLED”
f. “hd.lib hdu{d}.lib” are included in Configuration Properties → Linker → Input → Additional Dependencies.

6. **CPNMouse**

a. Download CPNMouse from: http://cpnmouse.sourceforge.net/
b. CPNMouse driver installation uses driver-binary-1.0.1.zip and instruction from http://cpnmouse.sourceforge.net/install.html
c. For development uses cpnmouse-api-0.9.3.zip, copy all include files which are “guid.h”, “hapi.h”, and “lapi.h” to “C:\Program Files\Microsoft Visual Studio 8\VC\include\cpnmouse” and “cpnmouse.lib” to “C:\Program Files\Microsoft Visual Studio 8\VC\lib”
d. For compiling GiPSiNet with CPNMouse enable, set Preprocessor Setting in Toolkit project with “_USE_CPNMOUSE”

7. Real-time CORBO with ACE\TAO v5.6

a. Download TAO from: http://www.cs.wustl.edu/~schmidt/TAO.html.

b. Install and compile the package.
Please follow the instruction from links below:
http://www.dre.vanderbilt.edu/~schmidt/DOC_ROOT/ACE/ACE-INSTALL.html
http://www.dre.vanderbilt.edu/~schmidt/DOC_ROOT/TAO/TAO-INSTALL.html

Installation summary:
For build ACE, create “config.h” at “$ACE_ROOT\ace” directory that contains:

```
#include "ace/config-win32.h"
```

The ACE/TAO needs some modification the source code due to definition conflict.

If the ACE/TAO version is 5.5 or below, modified the following:

```
ACE_wrappers\ace\os_include\sys\os_types.h (no need to change in new tao)
      #if !defined(__MINGW32__)
            typedef unsigned int mode_t;
      #endif /* !__MINGW32__ */
```

If the ACE/TAO version is 5.6 or below, modified the following:

```
ACE_wrappers\ace\os_include\sys\os_types.h
      #if defined (ACE_WIN32) && !defined(__MINGW32__)
            typedef int pid_t;
```

If the ACE/TAO version is 5.6.6 or higher, no need to modify.

c. Add Environment Variables

- ACE_ROOT	“C:\ACE_wrappers”
- TAO_ROOT	“C:\ACE_wrappers\TAO”
- PATH	“%PATH%;C:\ACE_wrappers\bin;C:\ACE_wrappers\lib”

8. GiPSiNet

a. Download and install Perl from http://www.perl.org

b. Run “GiPSiNet.bat” in directory “GiPSiNet”. This will generate the TAO related files for all three TAO services, which are GiPSiNet_HM, GiPSiNet_SK, and GiPSiNet_VE.

c. Open the “GiPSiNet.sln”, clean the solution, then compile and build it.

d. Running NameService in “C:\ACE_wrappers\TAO\orbsvcs\Naming_Service\” directory as the following

```
Naming_Service -ORBENDPoint iiop://129.22.151.97:1234567

Note: IP address and port can change.
```

e. Running GiPSiNet options

For start GiPSi Application using:

```
      application “xml_project_file”
```

For start GiPSiNet application using:

```
      application “xml_project_file” -net “options” “nameService”
```

```
where “options” are the following:
      startServer       = start GiPSiNet Server
      startClient       = start GiPSiNet Client which start HM and VE
      startHapticClient = start GiPSiNet Haptic Client
      startVEClient     = start GiPSiNet Visualization Client
and “nameService” is the TAO name service ip and port which is the same
as NameService (iiop:129.22.151.97:1234567). The “xml_project_file” is
not need for start server.
```

For examples:

runGiPSiNetServer.bat file:

```
cd GiPSi\projects\endo
..\..\..\Application\Release\Application -net startServer iiop:129.22.151.97:1234567
```

runGiPSiNetClient.bat file:

```
cd GiPSi\projects\endo
..\..\..\Application\Release\Application endo_haptic_camera.xml -net startClient
iiop:129.22.151.97:1234567
```

9. Visual Studio Settings

a. Start Visual Studio

b. Tools→Options→Projects and Solutions→VC++ Directories

c. Select Include Files,

Add “C:\Program Files\Intel\MKL\8.0\include”
Add “C:\Program Files\SensAble\3DTouch\include”
Add “C:\Program Files\SensAble\3DTouch\utilities\include”
Add “$(ACE_ROOT)”
Add “$(TAO_ROOT)”
Add “$(TAO_ROOT)\orbsvcs”

d. Select Library Files,

Add “C:\Program Files\Intel\MKL\8.0\ia32\lib”
Add “C:\Program Files\SensAble\3DTouch\lib”
Add “C:\Program Files\SensAble\3DTouch\utilities\lib”
Add “$(ACE_ROOT)\lib”

e. Select Executeable Files,

Add “$(ACE_ROOT)\bin”

10. GiPSiNet Compiler Directive

If you want to compile with the network functionality, add “GIPSINET” to Preprocessor Definitions to Application and Toolkit projects

Other Notes:

Make sure that in GiPSi Solution Properties, the following settings are used for “All Configurations”:

1. “ALGEBRA_USE_MKL” and “WIN32” are defined in Configuration Properties → C/C++ → Preprocessor → Preprocessor Definitions.
2. “..\Common” is included in Configuration Properties → C/C++ → General → Additional Include Directories.
3. “mkl_c.lib pthreadVC1.lib glut32.lib glu32.lib opengl32.lib odbc32.lib odbccp32.lib” are included in Configuration Properties → Linker → Input → Additional Dependencies.

The .XML Project Directory Structure

The main program is in “application\release” directory. The default xml project directories are the following:

- The xml projects files are in “\GiPSi\projects\”
- Each xml project has individual directory for example
  a. demo project is in “\projects\demo\”
  b. msd project is in “\projects\msd\”
  c. endo project is in “\projects\endo\”
- Each xml project has own “objects” and “textures” directory, for example for demo project directory contains
  a. Project file is in “\demo\”
  b. Object files are in “\demo\objects\”
  c. Texture files are in “\demo\textures\”
- The shader files are shared for all project in “\GiPSi\projects\Shaders\”

```
\Projects
    |------ \Shaders
    |
    |------ \demo
    |          |--- \objects
    |          |--- \textures
    |
    |------ \endo
               |--- \objects
               |--- \textures
```

Example for project command setup

In Project Property Pages, set the parameters as the following:

GiPSi Application

Configuration Properties → Debugging

→ Command Parameters: `Project.xml`

→ Working Directory: `..\GiPSi\projects\demo`

GiPSiNet Application

Configuration Properties → Debugging

→ Command Parameters: `Project.xml –net startServer iiop:129.22.151.97:1234567`

→ Working Directory: `..\GiPSi\projects\demo`

Example for project command line

GiPSi Application

```
cd GiPSi\projects\demo
..\..\..\Application\Release\Application Project.xml
```

GiPSiNet Application

```
cd GiPSi\projects\endo
..\..\..\Application\Release\Application -net startServer iiop:129.22.151.97:1234567
```

```
cd GiPSi\projects\endo
..\..\..\Application\Release\Application endo_haptic_camera.xml -net startClient
iiop:129.22.151.97:1234567
```