# SANKHYA Varadhi

Steps to be used to build Varadhi applications in VC++ (msdev) environment :

1. Set VC++ environment ( i.e run vcvars32.bat ) ( eg :
   C:\Program Files\Microsoft Visual Studio\VC98\BIN>
   vcvars32.bat
   The above VC installation may differ ).

2. Set Varadhi Environment
   C:\sankhya\varadhi> varadhi.cmd
   ( Make sure that the correct/required version of Varadhi you are setting)

3. Open your VC++ project file.
   %work_dir%> msdev my_project.dsw

4. Add required IDL file to "Source Files" of project workspace. Also add corresponding stub, skeleton source files (.cpp files) under "Source Files" and stub, skeleton Header files into "Header Files" of project workspace.

5. Now we need to set Custom build environment for IDL file. Select the IDL ( eg test.idl) and open open "Project Settings" or press "Alt F7" - Select "General" Tab and enable "Always use custom build step" - Now select "Custom Build". In the Commands list box, give command as below
   idlc ..\src\idl\test.idl
   ( The "..\src\idl\test.idl" is depends on your IDL file location )
   In the "Outputs" listbox add the follwing files one by one
   test_st.h, test_sk.h, test_st.cpp, test_sk.cpp

6. Open "Tools -> Options..." . In "Options" dialog box, select "Directories" tab. Select "Executable files" from "Show directories for". Now add Varadhi bin location ( in our case its "C:\sankhya\varadhi\bin\win32" ) in the "Directories" list box. Similarly select "Library files" and add Varadhi platform library location ( C:\SANKHYA\VARADHI\platforms\default-win32\lib ). It is for location of varadhi.lib. This can be added using /libpath option in "Project Settings" dialog. If we use custom build varadhi platform, use appropriate library location.

7. Open "Project -> Settings" or Press "Alt + F7" ( Make sure that when we open Project Settings that the project as the the current selection. )

8. Select "C/C++" Tab. Here, we need to add the required Preprocessor Definitions and Include Directories. In "Preprocessor definitions:" add any Varadhi specific definitions (like ENABLE_CPP_EXCEPTIONS etc..). In "Additional include directories:" add Varadhi include files location. Just add "$(VARADHI)\include". ( Here add is nothing but just appended the specified entry with comma seperation )

9. Next, select "Link" tab and add required varadhi platorm libraries name ( i.e varadhi.lib ). Also make sure that the system libraries netapi32.lib and ws2_32.lib is present.

10. Now export the makefile using "Project -> Export Makefile..." Here enable "Write dependencies..." and press "Select All" and give "OK"

11. Now you can build your project either using your exported makefile or from the workspace itself (using Build menu)

**Note:** Do all the settings in "Release" or "Debug" configuration. When one configuration is done, update the other configuration in same way.