Building NLMs Using Watcom's IDE v10.6 and C

ADAM JEROME
Developer Support Engineer
Developer Support

This DevNote outlines a method of building NLMs using Watcom's IDE v10.6. This method is recommended and approved by Novell Developer Support. Note that this method is specific to the C environment and is not suitable for the C++ environment.

Installation Steps

1. Install the Watcom development environment. Include `Novell NLM` as a *target platform*.

2. Install Novell's SDK CD. Include `NWSDK` in the installation process.

3. Modify your computer's path statement to include the `SDKCD9\NWSDK\TOOLS` directory.

   Note: This must be done so that:

   - Watcom's Linker will find Novell's `MAKEINIT` file.
   - Novell's NLM tools (such as `MSGLIB.EXE`) will be accessible.

4. Remove all files from the `WATCOM\NOVI` directory. Copy all .IMP files from the `SDKCD9\NWSDK\IMPORTS` to the `WATCOM\NOVI` directory. Example:

   ```
   COPY c:\sdkcd9\nwsdk\imports\*.imp c:\watcom\novi
   ```

   Note: The .imp files shipped with the Watcom 10.6 compiler are outdated. There is no known method of changing the IDE's path to CLIB.IMP. Therefore, the only alternative is to update the files in the location that the IDE expects to find them.

Build a New NLM Project (Tutorial)

1. Create a directory for your project. (Example: `C:\projects\hello`)

2. Activate the Watcom IDE.

3. Select "File/New Project" from the menu bar.

4. Specify the project directory and a project name. (Example: HELLO.wpj)

5. A *New Target* window is displayed allowing you to specify a *target name* and an *image type.* (By default they should be set similar to the following: Target name = "hello", Image type = "NLM [.nlm]"). Press the *OK* button. A *Source files* window will appear.

6. Initially, the *Source Files* window is empty. Press <Insert>. An *Add File(s) to ...* window will appear. Enter the name of the projects first C source file in the *File Name* field. (Example: HELLO.C). You may add other C source files as needed.

7. Continue using the *Add File(s) to* window to specify the location of Novell's PRELUDE.OBJ file. Do this by first setting the *List Files of Type* field to *All Files [*.*]*, then set the *Drives* field and *Directories* field to the `SDKCD9\NWSDK\IMPORTS` directory. Specify the file `PRELUDE.OBJ` and add it to the project. *Close* the *Add File(s) to* window.

8. Select *Options / Link for NetWare Switches...* from the menu bar. Select *2. Import, Export and Library Switches* from the scroll bar. *Check* the box labeled *No default libraries [op nod]*.

Specify .IMP files as needed in the *Import files(,):[imp]* field. (Example: For the “Hello world” example, you will have to specify the following in the *Import files(,):[imp]* field:

```
@$(nlm386imp)\threads.imp,@$(nlm386imp)\nlmlib.imp
```

Note: The macro value nlm386imp is defined in Novell’s makeinit file, in the `SDKCD9\TOOLS` directory. The actual .imp file names (ie: threads, nlmlib, etc.) may vary depending on the SDK release you have installed. This article assumes that you have installed SDK CD Release 9.

Press the *OK* button.

9. Select *Options / C Compiler Switches...* from the menu bar. By default, *1. File Option Switches* will be selected on the scroll bar. Set the *Include directories:[-i]* field to *$(nlm386hdr)*.

   Note: The macro value nlm386hdr is defined in Novell’s makeinit file, in the SDKCD9\TOOLS directory.

   An *IDE Request* window may appear asking *Mark all .c files in ‘...’ for remake?* If so, press the *Yes* button.

10. You may now wish to edit one (or more) of the previously specified C files. You may do so by double clicking on one of the files listed *Source files:* window. (Example: hello.c [n/a]). The following is a simple “Hello world” example:

```
#include <stdlib.h>
          #include <stdio.h>

          void main(void)
             {
             printf(“Hello world.\n”);
             return;
             }
```

11. Add the following function to one of your C source files (for example, add it to HELLO.C):

```
void __WATCOM_Prelude(void)
             {
             return;
             }
```

Note: The Watcom 10.6 compiler constructs OBJ files from C source files. When the Watcom compiler is invoked from the IDE, a reference is automatically implied to an external **__WATCOM_Prelude** symbol. There is no known method of disabling this behavior from the IDE (even though the reference is not needed or even referenced in the C source). Therefore, your application must provide its own reference as a “shim” or “stub.” This function will not be called by Novell’s PRELUDE.OBJ or Dynamic Linked Library NLMs.

(The example program HELLO.C will look something like this:)

```
          #include <stdlib.h>
          #include <stdio.h>

          void __WATCOM_Prelude(void)
                  {
                  return;
                  }

          void main(void)
                  {
                  printf("Hello world.\n");
                  return;
                  }
```

(Save the file, then proceed to step 12.)

12. Select *Targets / Make* from the menu bar. If all went well, you will find a powerful and highly efficient NLM in the target directory.