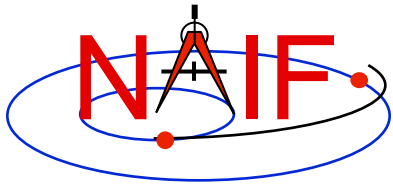


Navigation and Ancillary Information Facility

Preparing for Programming Using the SPICE Toolkit

March 2010



Setting Path to Toolkit Executables

Navigation and Ancillary Information Facility

Recommended for all languages

- **Unix**

- **csh, tcsh:** Use the set command to add the location of toolkit executables to your path.

- » `set path = ($path /my_directory/toolkit/exe)`
 - » `set path = ($path /my_directory/cspice/exe)`
 - » `set path = ($path /my_directory/icy/exe)`
 - » `set path = ($path /my_directory/mice/exe)`

- **bash**

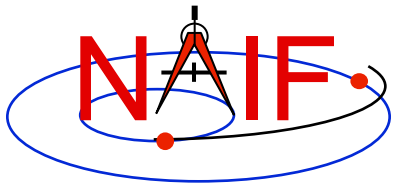
- » `PATH=$PATH:/my_directory/toolkit/exe`
 - » `PATH=$PATH:/my_directory/cspice/exe`
 - » `PATH=$PATH:/my_directory/icy/exe`
 - » `PATH=$PATH:/my_directory/mice/exe`

- **Windows**

- **Add location of toolkit executables to the environment variable PATH from the Advanced pane on the System Control Panel (Control Panel->System->Advanced).**

- » `drive: \my_directory\toolkit\exe`
 - » `drive: \my_directory\cspice\exe`
 - » `drive: \my_directory\icy\exe`
 - » `drive: \my_directory\mice\exe`

Replace the *italics* with the path in which you installed the toolkit on your computer.



Unix/Linux: Build

Navigation and Ancillary Information Facility

- **Compile and link an application, say *program*, against the SPICELIB/CSPICE libraries**
 - Assume SPICE is installed at `/naif/toolkit/` or CSPICE is installed at `/naif/cspice/`

» **C**

```
$ gcc program.c -I/naif/cspice/include /naif/cspice/lib/cspice.a -lm
```

» **FORTRAN**

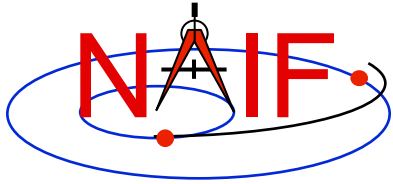
```
$ g77 program.f /naif/toolkit/spicelib.a
```

» **Some FORTRAN compilers (e.g. Absoft) require an additional flag "-lU77" to pull in the standard Unix symbols when linking against SPICELIB.**

- **The default SPICE library names do not conform to the UNIX convention `libname.a`. So you cannot use the library path/name options**

```
... -L/path_to_libs/ -lname
```

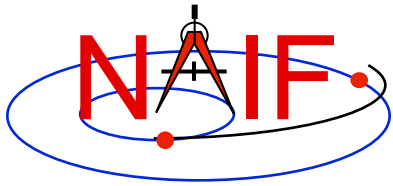
unless you rename the SPICE library.



Windows: Compiler settings

Navigation and Ancillary Information Facility

- **The standard installation of Microsoft Visual Studio may not update environment variables needed to use the C compiler (cl) from the standard DOS shell.**
 - You can set the environment variables by executing from a DOS shell one of the “vars32” batch scripts supplied with Microsoft compilers:
 - » `vars32.bat`
 - » `vcvars32.bat`
 - » `vsvars32.bat`
 - If available on your system, you can execute the “Visual Studio Command Prompt” utility from the *Programs -> Microsoft Visual Studio -> Visual Studio Tools* menu. The utility spawns a DOS shell set with the appropriate environment variables.



Windows: Builds

Navigation and Ancillary Information Facility

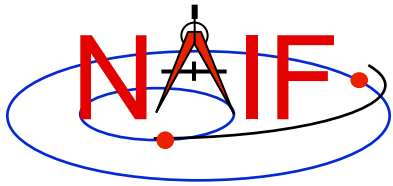
- Assume SPICE is installed at `C:\naif\toolkit\` with CSPICE installed at `C:\naif\cspice\`
 - Compile and link an application, say *program*, against the SPICELIB/CSPICE libraries

» C

```
> cl program.c -IC:\naif\cspice\include C:\naif\cspice\lib\cspice.lib
```

» FORTRAN

```
> df program.f C:\naif\toolkit\lib\SPICELIB.LIB
```



Icy: Register the Icy DLM to IDL (1)

Navigation and Ancillary Information Facility

Required for “Icy”

- **Unix and Windows**

- Use the IDL register command:

```
IDL> dlm_register, '_path_to_directory_containing_icy.dlm_'
```

```
IDL > dlm_register, '/naif/icy/lib/icy.dlm'
```

- Or, copy **icy.dlm** and **icy.so (icy.dll)** to IDL's binary directory

```
{The IDL install directory}/bin/bin.user_architecture
```

```
» /usr/local/itt/idl64/bin/bin.linux.x86/
```

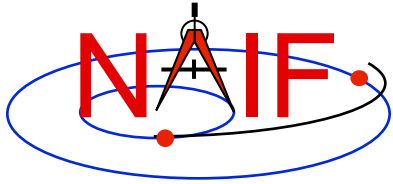
```
» C:\ITT\IDL64\bin\bin.x86\
```

- **Unix specific:**

- Start the IDL application from a shell in the directory containing both **icy.dlm** and **icy.so**.
- Append the path to your **icy.dlm** to the **IDL_DLM_PATH** environment variable to include the directory containing **icy.dlm** and **icy.so**, e.g.:

```
setenv IDL_DLM_PATH "<IDL_DEFAULT>:_path_to_directory_containing_icy.dlm_"
```

Caveat: with regards to the Icy source directory, *icy/src/icy*, do not invoke IDL from the directory, do not register the directory, and do not append to **IDL_DLM_PATH** the directory. This directory contains an “icy.dlm” but no “icy.so.”



Icy: Register the Icy DLM to IDL (2)

Navigation and Ancillary Information Facility

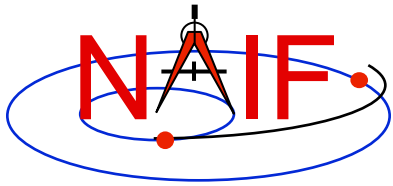
- **Windows specific:**
 - Set environment variable `IDL_DLM_PATH` from the *Advanced* pane of the *System Control Panel*.
- **Confirm IDL recognizes and can access Icy.**
 - Using the help command:

```
IDL> help, 'icy', /DLM
**ICY - IDL/CSPIICE interface from JPL/NAIF (not loaded)
```

» Appearance of the words “not loaded” might suggest something is wrong, but this is expected state until you execute an Icy command.

- **Execute a trivial Icy command:**

```
IDL> print, cspice_icy('version')
% Loaded DLM: ICY.
Icy 1.4.20 25-DEC-2008 (EDW)
```

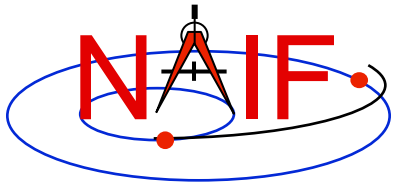


Icy: Using the IDL IDE

Navigation and Ancillary Information Facility

Recommended for “Icy”

- Use the IDL IDE’s preferences panel to set the current working directory to the location where you will be developing your lessons’ code.
- **Optional: Place your `dln_register` command in a start up script. Specify the script using the IDL IDE’s preferences panel.**



Mice

Navigation and Ancillary Information Facility

Required for “Mice”

- **Assume Mice is installed at `C:\naif\mice\` on Windows, or `/naif/mice/` on Unix/Linux. Use of Mice from MATLAB requires the Mice source and library directories exist in the MATLAB search path.**

- **On Windows:**

```
>> addpath('C:\naif\mice\lib')  
>> addpath('C:\naif\mice\src\mice')
```

- **On Unix/Linux:**

```
>> addpath('/naif/mice/lib')  
>> addpath('/naif/mice/src/mice')
```