Preparing for Programming Using the SPICE Toolkit

January 2009

Setting Path to Toolkit Executables

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

- **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/csupport.a \
                                       /naif/cspice/lib/cspice.a -lm
```

    - **FORTRAN**

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

    - **Some FORTRAN compilers (e.g. Absoft) require an additional flag "**`-lU77`**" when linking against SPICELIB to pull in the standard Unix symbols**
- **The default SPICE library names do not conform to the UNIX convention** `lib`*`name`*`.a.` **So you cannot use the library path/name options** `-L/`*`path_to_libs`*`/` **and** `-l`*`name`* **unless you rename the SPICE library.**

Windows: Compiler settings

- **The standard installation of Microsoft Visual Studio or Visual Toolkit may not update environment variables needed to use the C compiler (cl) from the standard DOS shell.**
  - **Example, environment variables for "cl" - Visual Studio 7:**
    - » `INCLUDE`
      - `C:\Program Files\Microsoft Visual Studio .NET\Vc7\include\`
      - `C:\Program Files\Microsoft Visual Studio .NET\FrameworkSDK\include\`
    - » `LIB`
      - `C:\Program Files\Microsoft Visual Studio .NET\Vc7\lib\`
      - `C:\Program Files\Microsoft Visual Studio .NET\FrameworkSDK\Lib\`
    - » `PATH`
      - `C:\Program Files\Microsoft Visual Studio .NET\Vc7\bin\`
  - **You can set the environment variables either by appending the directory paths shown above to the corresponding environment variable in the *Advanced* pane of the *System* Control Panel or by executing one of the “vars32” batch scripts supplied with the Microsoft compiler:**
    - » **vars32.bat**
    - » **vcvars32.bat**
    - » **vsvars32.bat**

Windows: Builds

- **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\csupport.lib \
                                        C:\naif\cspice\lib\cspice.lib
```

    - **FORTRAN**

```
> df program.f C:\naif\toolkit\lib\SUPPORT.LIB \
               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 icy.dlm>
    ```

  - copy icy.dlm and icy.so (icy.dll) to IDL's binary directory {The IDL install directory}/bin/bin.<your_arch>, e.g.:
    - » /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 icy.dlm directory>"
      ```
- Windows specific:
  - » Set environment variable IDL_DLM_PATH as described above from the *Advanced* pane of the *System* Control Panel.

Icy: Register the Icy DLM to IDL (2)

- **Confirm IDL recognizes and can access Icy**
  - **Using the help command:**

    ```
    IDL> help, ‘icy’, /DLM
    **ICY - IDL/CSPICE 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

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 `dlm_register` command in a start up script. Specify the script using the IDL IDE’s preferences panel.

Mice

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')
    ```