

Migrating Pre-UIM/X 3.0 Applications to UIM/X 3.0: Steps for Migrating .i files

Valerie J. Briggs- revised 4/22/02 and 1/27/06

These notes are provided as a general guide for someone who wishes to migrate .i files from pre-UIM/X 3.0 applications to UIM/X 3.0. It discusses two different situations: when UIM/X 3.0 is installed in Full mode, and when UIM/X 3.0 is installed in C-only mode. If you have further questions, please email ICS at info@ics.com or call 617.621.0060.

When the current UIM/X 3.0 distribution is installed in Full (C and C++) mode:

1. Move all your project files (.i, .prj, .op, .c, .h, etc.) to a new working directory.
2. Does your 2.9 project contain external swidget declarations? In UIM/X 3.0, a C++ wrapper class is defined for each Motif widget by default. Motif objects within an interface are declared as objects of these classes, rather than swidgets, providing C++ bindings for these objects. This could cause problems with external swidget declarations from projects built with UIM/X versions prior to 3.0, where the tool expects a class declaration rather than a swidget declaration.

To get around this problem, invoke UIM/X without the C++ bindings enabled:

```
uimx -xrm "Uimx3_0.UxNoUxBindings.set: true"
```

3. Instead of loading the .prj file, load the .i files individually. This will make it easier to resolve any interpreter errors or warnings. Keep in mind:
 - If an interface pops-up another interface, the interface containing the create function must be loaded first.
 - Load all component interfaces before any instance interfaces (the component contains the create function).
4. If step 2 applies to your project - external swidget declarations in 2.9 interfaces - do the following:

- Replace the declarations of external swidgets with a block of code that correctly declares the swidget. For example, if your project includes bulletinBoard1 and bulletinBoard2, where bulletinBoard2 is declared as global and you want to communicate with bulletinBoard2 from bulletinBoard1, do the following:

- Find the declaration in bulletinBoard1, and replace

```
extern swidget bulletinBoard2;
```

with the following:

```
#if defined(_cplusplus) && !defined(XT_CODE) && \
    !defined(NO_UXBINDINGS)
```

```
    extern UxBulletinBoard bulletinBoard2;
#else
    extern swidget bulletinBoard2;
#endif
```

- Apply the change in your editor.

5. When you have successfully loaded all the .i files associated with your project, save the project to a new 3.0 .prj file, and then regenerate your project code. This will update your main and make files with the correct 3.0 references.

- From UIM/X Main Menu:
 - Select File->Save Project As:
Enter a name with the .prj extension
Example: In the textField
Enter the project name:
/UpdateDirectory/Updated.prj

- Select File->Generate Project Code As:
With the following options enabled:

Menu Option	toggle state
-----	-----
Write All Interfaces	enabled
Write Main Program	enabled
Write Makefile	enabled

- You will notice that your project now:
 - Contains references to new Main and Make files
 - .i files contain updated header information.

Example:
! UIMX ascii 3.0C++ key: 1712

6. The project can now be loaded into UIM/X 3.0 with the C++ bindings enabled.

When the current UIM/X 3.0 distribution is installed in C-only mode:

1. Move all your project files (.i, .prj, .op, .c, .h, etc.) to a new working directory.
2. Instead of loading the .prj file, load the .i files individually. This will make it easier to resolve any interpreter errors or warnings. Keep in mind:
 - If an interface pops-up another interface, the interface containing the create function must be loaded first.
 - Load all component interfaces before any instance interfaces (the component contains the create function).
 - Errors common to the above are "Undefined or Unresolved Symbols"
3. When you have successfully loaded all the .i files associated with your project, save the project to a new 3.0 .prj file, and then regenerate your project code. This will update your main and make files with the correct 3.0 references.
 - From UIM//X Main Menu:
 - Select File->Save Project As:
Enter a name with the .prj extension
Example: In the textField enter the project name:

/UpdateDirectory/Updated.prj

- Select File->Generate Project Code As:
With the following options enabled:

Menu Option	toggle state
-----	-----
Write All Interfaces	enabled
Write Main Program	enabled
Write Makefile	enabled

- You will notice that your project now:
 - Contains references to new Main and Make files.
 - .i files contain updated header information.Example:

! UIMX ascii 3.0C++ key: 1712