Scripting Mac F2C

Introduction and Quick-Start

Mac F2C is AppleEvent savvy. It supports drag-&-drop translation. More importantly, you can drive Mac F2C translations entirely using AppleEvents.

The AppleEvent implementation in Mac F2C v1.3 is based on Apple's Object Model and utilizes a subset of the Core Suite. Events from the Core Suite are used to get and set the various options available in Mac F2C. For detailed information on the events in the Core Suite refer to the AppleEvent Registry. Mac F2C also supports a custom "FORTRAN-to-C Suite" that provides two different AppleEvents for translating FORTRAN to C:

• The first custom AppleEvent (the f2c AppleEvent) accepts a list of files and allows you to pass option choices to the translator in a UNIX-style option string (see the chapter "UNIX f2c Option Switch Syntax" for more information on the UNIX-style option switches). The f2c AppleEvent ignores any option settings specified in the FORTRAN Options, C Options, Prototyping Options, Warning Options, and Advanced Options dialogs (it does, however, use the settings in the Compiler Interactions and Mac F2C Preferences dialogs).

• The second AppleEvent (the translate AppleEvent) only accepts a list of files. It uses the all of the options settings specified in the various Options dialogs. You can use the get/set AppleEvents (from the Core Suite) to set the options as desired prior to sending a translate AppleEvent.

Which of these two AppleEvents you use is purely a matter of personal choice.

Some sample AppleScripts are included as part of the Mac F2C distribution package, in the Mac F2C Extras folder. They were written using v1.1 of the AppleScript™ extension and v1.1 of the AppleScript Editor. These scripts illustrate how to use the various events to control Mac F2C. The easiest way to drive Mac F2C from your own scripts is simply to copy and paste appropriate portions of the sample scripts into yours andthenmodify them as required. Mac F2C also includes a full aete resource, so you can get detailed information on the exact syntax of the AppleEvents supported by Mac F2C from its dictionary.

Changes from Mac F2C v1.2

Mac F2C now supports Apple's Object Model and implements a subset of the Core Suite.

The translate AppleEvent no longer accepts optional parameters specifying the value of the various option settings. Each option now corresponds to a property of the application and can be accessed using the Core Suite get and set AppleEvents. Refer to the dictionary for the names of the properties.

THE DETAILS

The remainder of this chapter provides more detailed information for users who wish to know more details about sending AppleEvents to Mac F2C. In particular, the information below has all the information needed to write code that sends AppleEvents to Mac F2C. If you simply want to write an AppleScripts to drive Mac F2C, refer instead to the scripting dictionary built into the Mac F2C application itself.

The constants appearing in this typeface below are defined for you in the file F2cAETerms.h, which is included in the Mac F2C Extras folder. You can #include this file in any C/C++ source file you write that sends AppleEvents to Mac F2C.

Classes

- cApplication:
  the application: one instance, the Mac F2C application

- cWindow:
  a Macintosh window: one instance, the status window

Containment Hierarchies

"null"	(the topmost object)
cApplication	(Mac F2C, synonymous with "null")

cWindow (translation status window [only 1 window])

Class Properties

• cApplication:

< FORTRAN Input Options >

pCheckSubscripts
    typeBoolean
    Check subscripts are within array bounds at run time.
pDoLoopsOnce
    typeBoolean
    Perform DO loops at least once if reached (FORTRAN-66 rules).
pDefaultVariableType
    typeEnumeration
    Default type of undeclared variables is:
        kUndefined - undefined (= error message)
        kFortranRules - default FORTRAN rules (= first letter I-M -> integer)
pFreeFormatInput
    typeBoolean
    Assume free-format input.
pCutAtColumn72
    typeBoolean
    Treat text appearing after column 72 as an error.
pDoubleComplex
    typeBoolean
    Implicitly recognize DOUBLE COMPLEX.
pBackslashEscapes
    typeBoolean
    Recognize backslash escapes in character strings.
pIntegerSize
    typeEnumeration
    Size of INTEGER & LOGICAL variables (INTEGER*4 always 4-bytes):
        k4Byte - 4 bytes
        k2Byte - 2 bytes
pInteger8
    typeBoolean
    Allow INTEGER*8 declarations.

< C Output Options >

pCodeGenerated
    typeEnumeration
    The variety of C code to generate:
        kKrC - K&R C code
        kAnsiC - ANSI C code
        kCpp - C++ code
pHonorCase
    typeBoolean
    Honor case of variable and external names.

pLocalVariables
typeEnumeration
Storage class for local variables:
kAutomatic - automatic (= stack) storage
kStatic - static (= like globals) storage

pAlignCharStrings
typeBoolean
Align char strings on words (FORTRAN-66 style Hollerith).

pCommonAsExtern
typeBoolean
Declare uninitialized COMMON to be "Extern".

pCommonSeparate
typeBoolean
Place uninitialized COMMON blocks in separate files.

pCommonLookLocal
typeBoolean
Make COMMON block members look like local variables.

pFortranEvaluation
typeBoolean
Use temp values to force FORTRAN order of evaluation.

pDoublePrecision
typeBoolean
Perform all floating-point operations in double precision.

pCastRealFunctions
typeBoolean
Cast values of REAL functions to REAL (including intrinsics).

pPromoteSingleToDouble
typeBoolean
Promote REAL to DOUBLE PRECISION, COMPLEX to DOUBLE COMPLEX.

pMultiDimSubscripts
typeBoolean
Preserve multidimensional subscripts.

< Prototype Generation Options >

pPrototypes
typeEnumeration
Controls generate prototypes:
kCodeOnly - code only, no prototypes
kCodeAndProtos - both code and prototypes
kProtosOnly - no code, only prototypes

pRetranslate
typeBoolean
Re-translate if prototypes or declarations may change.

pInferTypes
typeBoolean
Infer the type of untyped EXTERNAL functions used only as parameters to previously defined procedures.

< Warnings Issued Options >

pSourceAsComments
typeBoolean
Include the original FORTRAN source as comments .

pLineNumbers
typeBoolean
Include the original FORTRAN line numbers in “#line” lines.

pRejectFortran77
typeBoolean
Reject extensions to the FORTRAN-77 standard

pWarnings
typeEnumeration
Suppress the following warnings (list permitted):
kNoWarnings - none
kF66Warnings - suppress FORTRAN-66 compatibility warnings
kAlignWarnings - suppress odd-alignment warnings caused by EQUIVALENCE/COMMON statements
kAllWarnings - suppress all warning messages

< Advanced Options >

pEquivalences
typeShortInteger
Maximum number of equivalences.

pExternals
typeShortInteger
Maximum number of external symbols.

pStatementLabels
typeShortInteger
Maximum number of statement labels.

pNesting
typeShortInteger
Maximum nesting of control structures.

pNames
typeShortInteger
Maximum number of names.

pLiterals
typeShortInteger
Maximum number of literal constants.

pContinuations
typeShortInteger
Maximum number of continuation lines.

pImpliedLabels
typeShortInteger
Maximum number of implied labels.

< Compiler Interaction Options >

pInteractWith
typeEnumeration
Which C compilers to interact with:
kCompilerNone - none
kCompilerTPM - THINK Project Manager (68k)
kCompilerSPM - Symantec Project Manager (PPC)
kCompilerCW68K - CodeWarrior 68K
kCompilerCWPPC - CodeWarrior PPC
kCompilerCWIde - CodeWarrior IDE (v1.3)

pInteractHow
typeEnumeration

What kind of interaction to have with C compilers:

```
kNoInteraction     - none
kTouchFiles        - touch files generated by Mac F2C
kCompileFiles      - compile files generated by Mac F2C
kUpdateProject     - update whole projects containing files
                       generated by Mac F2C
kBuildTarget       - build targets of projects containing files
                       generated by Mac F2C
```

pInteractAddFiles

typeBoolean

Add files generated by Mac F2C to project files as required.

< Mac F2C Preference Options >

pSkipNonFORTRAN

typeBoolean

Silently skip non-FORTRAN files.

pOutputFileType

typeType

Creator type for output files.

• cWindow:

<none>

No properties implemented.

Object Specification

• cApplication:

formName "\pMac F2C"

• cWindow:

formAbsolutePosition 1 (there is only one window)

Events

Required Suite

Mac F2C supports the four required Apple Events: open application, open documents, print documents, and quit application. These events behave as follows:

- open <application>: No unusual actions are performed.

- open <documents>: Documents listed are translated into C using the default options settings. An error message is issued if the document name does not end in “.f” or “.F”. An open event is equivalent to a translate event (see below) with no optional parameters.

- print <documents>: Silently ignored; Mac F2C does not print any documents.

- quit: Quits Mac F2C.

Note that Mac F2C does not own any files (i.e., there are no files of type Mac F2C except for the application itself). The open event is used to support drag-&-drop operations. This allows you to translate a bunch of FORTRAN files by dragging them and dropping them on top of Mac F2C. The print event is supported to allow printing of the status window, NOT printing of source code or other documents.

More detailed descriptions of the required events follow:

- open <application>

Start the application with no documents to pass
(Mac F2C starts and waits for user commands)
Class: kCoreEventClass
ID: kAEOpenApplication
Parameters: <none>

- open <document>

Open a list of documents
(Mac F2C will translate each one using current option settings)
Class: kCoreEventClass
ID: kAEOpenDocuments
Parameters:
keyDirectObject

required
typeAlias
List of documents to open

- print <document>

Print a list of documents
(Mac F2C ignores this AppleEvent when passed documents)
Class: kCoreEventClass
ID: kAEPrintDocuments
Parameters:

```
    keyDirectObject
        required
        typeAlias
        List of documents to print
```

• quit
Quit the application
Class: kCoreEventClass
ID: kAEQuitApplication
Parameters: <none>

Core Suite Subset

Mac F2C supports several events from the Core Suite. Most important of these are the get and set events. Use these events to examine and change the value of the various properties that correspond to the Mac F2C options. Detailed descriptions of these events follow:

• save <window>
Save the translation status window contents
Class: kAECoreSuite
ID: kAESave
Parameters:

```
    keyDirectObject
        required
        cObjectSpecifier
        Object to be saved
```

• print <window>
Print the translation status window contents
Class: kCoreEventClass
ID: kAEPrint
Parameters:

```
    keyDirectObject
        required
        cObjectSpecifier
        Object to be printed
```

• get <property>
Get a property
Class: kAECoreSuite
ID: kAEGetData
Parameters:

```
    keyDirectObject
        required
        cObjectSpecifier
        Object whose data is to be returned
    keyAERequestedType
        optional
        cType
        List of preferred returned data types
Reply parameters:
     keyAEResult
        typeWildCard
        The data requested
```

• set <property>

```
Set a property
Class:                  kAECoreSuite
ID:                       kAESetData
Parameters:
     keyDirectObject
        required
        cObjectSpecifier
        Object to be changed
     keyAEData
        required
        typeWildCard
        The new value
```

FORTRAN-to-C Suite

The FORTRAN-to-C Suite is a set of AppleEvents that can be used to control the translation of FORTRAN files to C files.   The FORTRAN-to-C Suite consists of two AppleEvents differing in how options are passed to the f2c kernel.

The first AppleEvent (the f2c event) mimics the operation of the original f2c program on UNIX machines.   In this AppleEvent options for the f2c kernel are passed as a single string of UNIX-style option settings.   This AppleEvent ignores any option settings specified in the FORTRAN Input, C Output, Prototype Generation, Warnings Issued, and Advanced Options dialogs (it does, however, use the settings in the Compiler Interface and Mac F2C Preferences dialogs). This means that if you do not include a UNIX-style option string, then NO option switches are passed to the f2c kernel.

The second AppleEvent (the translate event) is more “Macintosh” in that it takes a list of files and translates them using the current settings of the options (which can be controlled using get/set events).

These two events are:

• f2c: Translate the specified files (one or more) from FORTRAN to C using a UNIX-style string of options.

- translate: Translate the specified files (one or more) from FORTRAN to C using the current option settings.

IMPORTANT NOTE: Both of these events return an error code (keyErrorNumber) only if Mac F2C could not process your AppleEvent. If Mac F2C was able to successfully process your AppleEvent, then the result (keyAEResult) will contain the return code from the f2c kernel (otherwise there will be no result). To be sure that Mac F2C correctly translated a file, you must (a) first make sure there were no errors processing your event and then (b) check that the result of the event is 0 (zero). This may sound a bit convoluted, but it is in fact the way things are supposed to work. And if you think about it a bit it actually makes a lot of sense.

More detailed descriptions of the FORTRAN-to-C events follow:

- f2c <FORTRAN files>

This event causes Mac F2C translate one or more FORTRAN files. The options for the f2c kernel are specified by a UNIX-style set of switches (e.g. "-A -P -w -f"; see the chapter on UNIX f2c Option Switch Syntax for more information on the switches and their meaning). The current option settings in the FORTRAN Input, C Output, Prototype Generation, Warnings Issued, and Advanced Options dialogs are ignored (however, the settings in the Compiler Interface and Mac F2C Preferences dialogs are used). This means that if you do not include a UNIX-style option string, then no option switches are passed to the f2c kernel.

```
Class:                    kF2CEventClass
ID:                         kAEf2cEvent
Parameters:
     keyDirectObject
        required
        typeAlias
        List of FORTRAN files to translate
     keyUnixCmdStr
        optional
        typeChar
        UNIX-style option string
Reply parameters:
     keyAEResult
        typeInteger
        List of the return codes from the f2c kernel, one per file translated
     keyErrorNumber
        typeLongInteger
        Error code if unable to process your event; not provided if no
        processing errors occur; compile results (including compile errors)
        are returned in keyAEResult.
     keyErrorString
        typeChar
         Error message whenever keyErrorNumber is non-zero
```

- translate <FORTRAN files>

This event will have Mac F2C translate a list of FORTRAN file. Current option settings are used. You can change the option settings by using the get and set AppleEvents.

Class: kF2CEventClass
ID: kAETranslateEvent
Parameters:
keyDirectObject
required
typeAlias
List of FORTRAN files to translate
Reply parameters:
keyAEResult
typeInteger
List of the return codes from the f2c kernel, one per file translated
keyErrorNumber
typeLongInteger
Error code if unable to process your event; not provided if no processing errors occur; compile results (including compile errors) are returned in keyAEResult.
keyErrorString
typeChar
Error message whenever keyErrorNumber is non-zero