Frontier SDK 2.1

UserLand Software, Inc.

UserLand Software is the developer of the UserLand Frontier scripting system. The company is located at 400 Seaport Court, Redwood City, CA 94063. 415-369-6600, 415-369-6618 (fax). UserLand, Frontier, Frontier Runtime and Frontier Extras are trademarks of UserLand Software, Inc. Other product names may be trademarks or registered trademarks of their owners.

Email: userland.dts@applelink.apple.com. If you're an AppleLink user, check out the UserLand Discussion Board under the Third Parties icon. CompuServe users enter GO USERLAND at any ! prompt. The UserLand Forum is in the Computing Support section on CompuServe. On America On-Line, enter the keyword USERLAND.

Comments, questions and suggestions are welcome!

Important Disclaimer

Please Read Before Using This Software

FRONTIER SDK IS PROVIDED ON AN "AS IS" BASIS. USERLAND SOFTWARE, INC. ("USI") EXCLUDES ANY AND ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL USI BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND. IN THE EVENT THE SOFTWARE IS DEFECTIVE YOUR SOLE REMEDY IS TO RETURN THE SOFTWARE TO USI FOR REPLACEMENT. IN ALL EVENTS, USI'S MAXIMUM LIABILITY SHALL BE LIMITED TO A REFUND OF THE PRICE PAID BY YOU FOR THIS SOFTWARE.

Requires Think C 6.0, Frontier 1.0, System 7.0

Frontier SDK 2.1 requires Macintosh System 7.0 or greater, since it builds on the Apple Event Manager.

All the toolkits and sample code are provided in a form compatible with Think C or Symantec C++ 6.0.

Frontier 1.0 or greater is required for testing with all the toolkits except IAC Tools.

The latest release of Frontier is version 2.0, shipped in October 1992. The 1.0 to 2.0 upgrade is free. If you're using Frontier 1.0, be sure to send in your registration card for an automatic free upgrade.

A bugfix patcher has been released by UserLand, upgrading Frontier 2.0 to 2.0.1. It can be downloaded from UserLand's support services on CompuServe, America On-Line or AppleLink.

If you include this code in your product, the features it supports will only be functional on Macintoshes running System 7.0 or greater.

Introduction

This collection of sample programs, source code and documentation is called the Frontier Software Developer's Kit or Frontier SDK.

Frontier SDK was developed by UserLand Software to help C and Pascal developers work with the interapplication communication capabilities of the Macintosh operating system and to take advantage of scripting software such as UserLand Frontier and Apple's AppleScript.

Included are libraries and sample programs that show you how to:

- Add scriptable Apple Event wires to an existing application; (IAC Tools)
- Create a new application with IAC capabilities; (Applet Toolkit)
- Support the menu sharing protocol; (Menu Sharing Toolkit)
- Send an IAC message to Frontier to run a short script. (Frontier Do-Script)
- Write a code extension to run in Frontier; (XCMDS & UCMDS)
- Exchange outline-structured lists between applications. (Outline Sharing Toolkit)

UserLand Software distributes Frontier SDK at no charge, with no royalty or license fee. The package includes full C source code to all libraries and sample programs.

If you've found Apple Events confusing or if it looks like too much work, we encourage you to explore the code in Frontier SDK. Perhaps it will clear up the confusion, or make IAC more approachable. Even if you don't use any of the source code in your development work, it can serve as a collection of solved problems that may fit into your development work at some time in the future.

This document provides an overview of how all the components of Frontier SDK tie together.

General Notes

We Use Think C 6.0

All of the libraries and sample code are provided in a format compatible with Symantec's Think C 6.0. Project files and .rsrc files are provided.

All projects can be compiled with the "Require Prototypes" flag turned on.

If you develop in an environment other than Think C, the sample code can be adapted to work in almost any Macintosh development environment.

Installing for Think C 6.0

Copy the Frontier SDK folder into the folder that contains the Think Project Manager application and debugger. It's a magic folder: you can #include headers in this package in your programs by enclosing them in <angle brackets>. The Think compiler will find them.

For example, if you add this line to the top of your program file, you can call menu sharing routines and access menu sharing data structures:

```
#include <menusharing.h>
```

This is an important requirement -- if the toolkits are not in the same folder as the Think Project Manager application, many of them will not compile.

No License Fee or Royalty

There is no license fee or royalty due on the use of any of the code in Frontier SDK. You may convert the code to run in other development environments. You may include the code in any shipping product. You may modify the code to fix bugs or extend its capabilities.

However, you must maintain our copyright notice on each file you use. You may not distribute modified versions of any of these programs in source or executable form without the written permission of UserLand Software. The intent behind this restriction is to maintain a standard, not in any way to limit the ways this code can be used.

Although it’s not a requirement, we ask that you send us one copy of any completed software that uses this SDK so we can test it with Frontier, and so we can learn from your use of IAC capabilities and Frontier-based scripting.

Testing & Refinement

If you follow the examples and documentation in Frontier SDK your product should work for Frontier script writers. But the ultimate test of compatibility will be trying out your interapplication capabilities using Frontier.

Add wires to your program thoughtfully. Think about scripts that people will want to write to customize and automate your software. Keep your scripting interface as simple as possible.

Most important: become a script writer yourself. It’s like anything else in software, if you don’t use it, it won’t be right.

Frontier Install Files

When you’re satisfied with your scripting interface, export a Frontier install file and make it available for script writers, either by including it with your software, or uploading it to one of UserLand’s on-line services.

We’ve included an install file for each of the sample applications in the SDK. When you double-click on these files, Frontier copies resources from the file into the Frontier.root object database file. Each install file contains a table of glue scripts and a shared menu bar.

It’s very easy to create a Frontier install file. See the “Frontier Install File Creator” folder in the Extras folder, for step-by-step instructions on creating a Frontier install file for your application.

Frontier Info

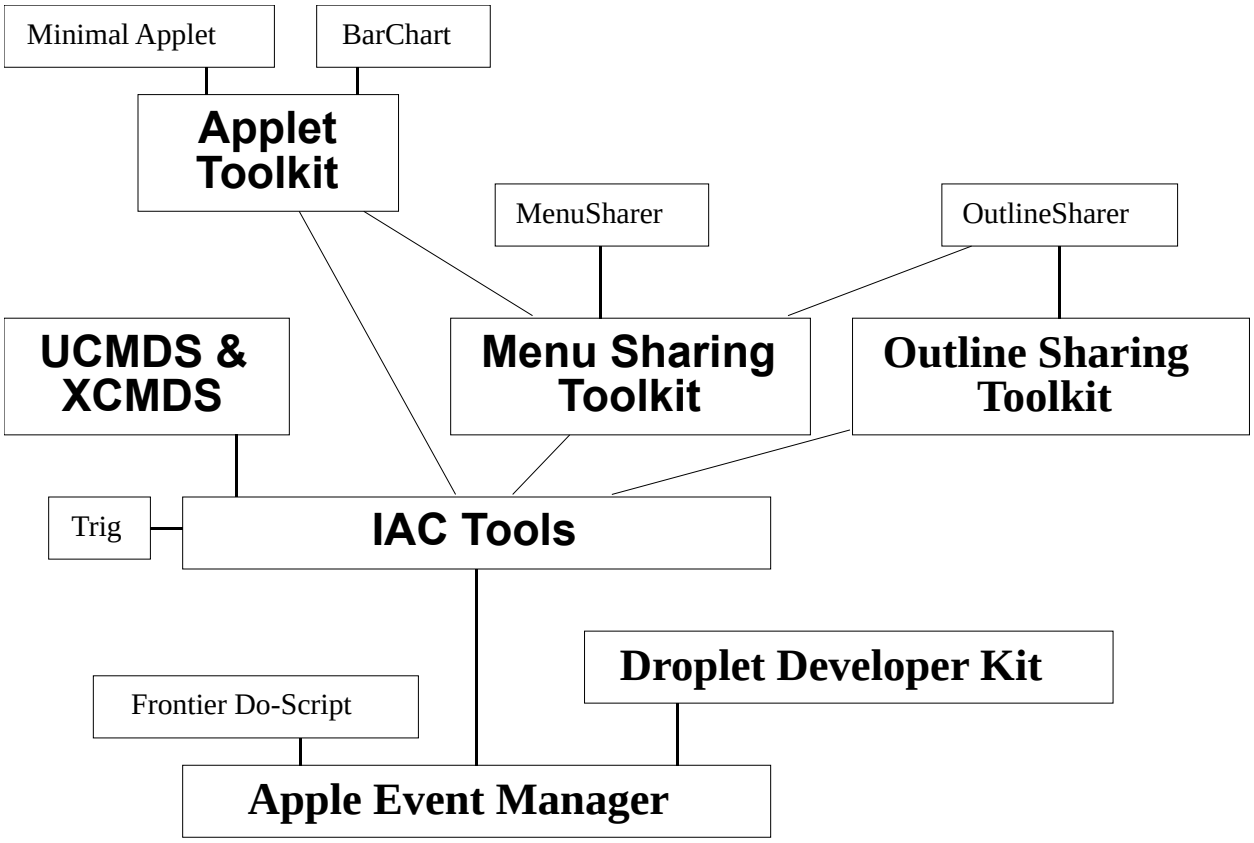
A tutorial in screen shots for Frontier 1.0 is available on CompuServe in Library 8 in file INFO.KIT. On AppleLink, the same file is available at the top level of the UserLand Discussion board.

Frontier Runtime

The Frontier Runtime package can be downloaded from CompuServe in Library 4, RUNT.SIT. On AppleLink it’s in the Hot Apps! section of the UserLand Discussion Board.

Schematic

Here's a schematic that illustrates how the toolkits and the sample code in Frontier SDK connect:



Toolkits

IAC Tools

The C source files in this folder provide a simplified API for sending and receiving Apple Events. A lot of the power of the Apple Event Manager is hidden by this API. But we've found that many applications of IAC don't require all the power of the Apple Event Manager.

Except for the Droplet Developer Kit, all the other libraries in Frontier SDK build on the IAC Tools API.

Menu Sharing Toolkit

Menu Sharing allows Frontier script writers to add menus and sub-menus to your application's menu bar.

Most of the sample programs included in Frontier SDK also support menu sharing; MenuSharer is the simplest sample app.

Important: In Frontier SDK 2.0, we changed our recommendation for where to call CheckSharedMenus. If you are using the 1.0 Menu Sharing Toolkit, please refer to the Change Notes section in the Menu Sharing Toolkit readme file.

Applet Toolkit

The Applet Toolkit makes it simple to build new double-clickable Macintosh applications that have strong support for Apple Events and menu sharing.

If you're developing a new application specifically to be integrated with other software, the Applet Toolkit is perfect.

UserLand's DocServer application is implemented using the Applet Toolkit, as are new utilities in development at UserLand.

Two sample applications are provided. Minimal Applet is a simple multi-window text editor, BarChart is a color charting program.

Because the Applet Toolkit is an application framework, certain Apple Event messages can be sent to all programs based on the Applet Toolkit.

Outline Sharing Toolkit

Outline sharing allows programs to exchange lists and lists-of-lists as part of Apple Event messages. Hierarchic lists come up all the time when designing software. Until outline sharing there was no standard way for applications to easily communicate and manage large hierarchic structures.

The Outline Sharing Toolkit also suggests how to approach toolkits for other data types that can be transported using the Apple Events protocol.

OutlineSharer is the sample application for outline sharing.

UserLand's "Bullet" application, a color outline browser, builds on the Outline Sharing Toolkit. It's available as a separate download.

UCMDs & XCMDs

Frontier 2.0 supports two kinds of code extensions: UCMDs and XCMDs.

In both models, you use a C or Pascal compiler to create a code resource, and then copy the code resource into Frontier's object database. Once the code is in the database, your scripts can call them, send parameters, and receive returned values, just as if they were verbs built into the UserTalk language.

XCMDs are HyperCard 1.0-compatible code extensions. UCMDs are compiled Apple Event handlers. Each approach has advantages. See the readme file in the UCMDS & XCMDS folder for details.

Included in the UCMDs & XCMDs folder are 4 sample XCMDs and 3 sample UCMDs.

Sample Code

Apple Events 101

Basic introduction to Apple Events programming using Think C. Two sample programs in source code: Client and Server. Client sends a series of Apple Events to Server. They are minimal programs, makes it easy to see how the Apple Events work.

BarChart

This is a simple demo application for the Applet Toolkit. It doesn't implement as many of the callbacks as Minimal Applet. It's different in that it uses color, and it's a graphics application, and therefore implements some of the callbacks that aren't implemented in Minimal Applet.

MenuSharer Program

This demo application shows you how to add menu sharing to your application.

It's the reference app for the the tutorial in the Menu Sharing Toolkit folder.

MenuSharer also illustrates how to add Apple Event handlers to your application.

Minimal Applet

This is the main demo application for the Applet Toolkit. It's a multi-window TextEdit-based editor.

It is also used as an example in the "Frontier Install File Creator" sub-folder in the Extras folder.

OutlineSharer

This is a demo application for the Outline Sharing Toolkit. It implements four outline-related Apple Events. A Frontier install file is provided for OutlineSharer. In the OutlineSharer.examples table there are several scripts which test out the Apple Events implemented in OutlineSharer.

Trig

This is a demo application for use of the IAC Tools library and system event handlers.

Frontier Do-Script

This is the simplest of the sample programs. It implements a routine called FrontierDoScript that shoots an IAC message at Frontier asking it to run a short script, and returns a string representation of the value of the script.

This program does not use the IAC Tools library. For a version of FrontierDoScript that builds on top of the IAC Tools library, check out appletfrontier.c in the Applet Toolkit folder.

Extras

Nerd's Guide to Frontier

If you've been wondering what Frontier is all about, read this document first.

It's a concise technical overview of Frontier 2.0 for experienced C/Pascal programmers. Screen shots, lots of sample scripts. From "Hello World" to object model scripting. Explains the UserTalk language, storage system, script editor/debugger, menu sharing, desktop scripts, droplets, DocServer.

Customizing Think C...

A case study in customizing Think C with Frontier scripts. It takes you thru the development process of a single script, starting with a first proof-of-concept version, all the way to a very useful and complete script that saves us time every day.

Frontier Install File Creator

You've added Apple Event support to your application. You've installed the Menu Sharing Toolkit. You're almost ready to ship.

But how are script writers going to learn about your Apple Event support?

This document is part cookbook, part style guide. If you follow it carefully, Frontier script writers will love you.

Droplet Developer Kit

The Droplet Developer Kit includes all the scripts and docs you need to create your own droplets — double-clickable applications that have an embedded script that is run once for each file, folder or disk icon that's dragged onto the app. It's small, they're easy to write, and lots of fun to run.

Runtime License Agreement

Frontier Runtime can be licensed by commercial software developers to be included with Apple Event-aware applications. The license fee is $100 per year, no royalties.

This is the license agreement for bundling Frontier Runtime.

Who Should Use What?

If you're adding wires to an already-existing application…

Check out the IAC Tools folder. You may find that the examples in iac.c make the lightbulb go off about the Apple Event Manager.

If you already have scriptable IAC wires…

Please create a Frontier install file for your application. Install files are a by-product of testing your IAC wires with Frontier. It's a simple process to create your install file. Check out "Frontier Install File Creator" in the Extras folder.

Menu Sharing is very low overhead and easy to implement; and your users and script writers will appreciate it.

If you're writing a new application…

Consider using the Applet Toolkit. It factors out of a lot of the common code that all multi-window Macintosh applications must implement. And it gives you menu sharing with Frontier for free. And it implements a set of Apple Event messages automatically, and makes it easy for you to add more.

If you develop using Think Class Library or MacApp…

It would be great to see "applet-like" capabilities implemented in the class hierarchies of MacApp and Think Class Library. Would you share them with other developers who use your environment? Please contact UserLand Software if you're interested.

Frontier SDK 2.1 -- 6/19/93 DW

Recompiled with Think C 6.0

All the source code in Frontier SDK 2.1 has been compiled with Think C 6.0. All the project files are 6.0 format.

Some minor source code changes were needed to make the SDK source code compatible with Think C 6.0. In all cases, code could be recompiled with added type casts. This was needed because MacHeaders now has function prototypes for all the Macintosh toolbox routines. The code got better as a result of these changes.

New files/folders in Frontier SDK 2.1

- Sample Code:Apple Events 101
- Extras:Nerd's Guide to Frontier
- Extras:Customizing Think C...

Reworked iacbinary.c, impact on Menu Sharing Toolkit

Generalized the routines in iacbinary.c so they could pass thru the binary type of a parameter. Minor change to calling of these routines in menusharing.c (in GetMenuHandles and GetSharedMenus).

Changed the prototypes in iac.h.

Bug in iacboolean.c

In IACpushbooleanparam, we were using sizeof (short) to determine the number of bytes to push onto the Apple Event record. It's equal to 2. But sizeof (Boolean) equals 1. So false wasn't always going back as 0! It was pushing garbage on the second byte. My stressBase script caught this one... DW

New data types for IAC Tools

New files: iacdate.c, iacfilespec.c, iaclist.c, iacrgb.c, iacrecord.c, iacstring4, iactable.c.

iacfilespec.c moves FSSpec records between apps, generally they're more accurate than strings. An upcoming release of the Applet Toolkit is entirely FSSpec based.

iaclist.c allows you to manage Apple Event lists. iacrecord.c manages Apple Event records.

> **Thanks to John Baxter for this code! He writes "You can get any of the supported types from a list or record, or put them into a list or record, or obtain information regarding the type or size of any AE parameter or any item in a list or record, or the keyword of the n-th item in a record. Note that an Apple event structure itself can be passed to the record-handling routines...in fact the implementation of all the tools has been changed to do just**

that, rather than duplicate the code. The Apple Event structure is simply a specialized AERecord structure."

iactable.c allows you to send and receive a packed Frontier 2.0 table. There are no routines to parse the contents of a packed table.

Changes to iacdouble.c

1. Added support for projects which are compiled with Think C's Universal format doubles, or with FPU instructions, or both.

All the Frontier SDK sample code is compiled with "Native Floating Point Format" on. See the Compiler Settings panel, in the Think C... command off the Options sub-menu of the Edit menu, in Think Project Manager.

2. Added a check which produces a compiler error (instead of incorrect results) if a project which includes iacdouble.c is compiled with the 8 byte doubles option. If either FPU code or Universal format doubles are used, the SANE library needs to be added to the project.
3. Changed the API for iacdouble.c functions. IACpushdoubleparam and IACreturndouble were inconsistent, in that one took the double on the stack and the other took a pointer to the double. Now, both take the double on the stack (pass by value).
4. Thanks to Mark Sundstrom and John Baxter for this code!

Trig application

- Changed calls to IACreturndouble (&x) to IACreturndouble (x) to match changes in iacdouble.c.
- Changed glue scripts that wire into Trig's system-level event handlers. Instead of calling Frontier's appleEvent built-in, we use the systemEvent built-in, which is more efficient and cleaner. It was introduced in Frontier 2.0, we must have missed this in the pre-2.0 review of Frontier SDK.
- Nice: this tests the changes to iacdouble.c. It seems to work even better now -- we get more precision from the computations.

Droplet Developer Kit

Added new routine in droplet.c -- getResourceAndDetach. It gets a resource and then detaches it. Changed all calls to GetResource to getResourceAndDetach.

This problem was pointed out on CompuServe (Bill Monk, I think). The old version was dangerous, but it probably never actually caused problems.