

IAC Tools 2.0

UserLand Software, Inc.

UserLand Software is located at 400 Seaport Court, Redwood City, CA 94063. 415-369-6600, 415-369-6618 (fax). UserLand and Frontier 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.

Comments, questions and suggestions are welcome!

Background

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

If you're going to do IAC you'll probably want to write routines that simplify it. Feel free to use our source code as a starting point, or use the IAC Tools API as provided in `iac.c`.

All the other libraries in Frontier SDK build on the IAC Tools API.

New in Frontier SDK 2.0: support for system event handlers and asynchronous Apple Events.

System Event Handlers

Background

In IAC Tools 2.0, we introduce a way to create system event handlers. These work exactly the same way as non-system Apple Event handlers, but they often run much faster. But there are limits to what you can do in a system event handler. Support for system event handlers is implemented in iacsysevents.c in the IAC Tools folder.

Frontier 2.0 has support for system event handlers, Frontier 1.0 does not. All future versions of Frontier will support system event handlers.

Installing a system event handler

Call IACinstallsystemhandler instead of IACinstallhandler. The parameter lists are identical, the handler works the same way, but the system handler installer does a lot more work for you both at install time and when the handler is invoked. For details, check out the source to IACinstallsystemhandler and callhandler in iacsysevents.c

Trig

Our sample application for system event handlers is Trig.

It installs a system event handler for all of its messages that can be handled without a full context switch. It installs a normal handler for messages that must be handled slowly, such as displaying a string in the message window.

You can explore the source code to see how system event handlers are implemented. Open trig.¹ in the Trig folder, then open main.c. You'll see two statements in function main ():

```
if (!IACinstallsystemhandler ('trig', typeWildcard, &handlefastverb))
    goto error;

if (!IACinstallhandler ('TRIG', typeWildcard, &handleslowverb))
    goto error;
```

You can view the source of IACinstallsystemhandler in iacsysevents.c and handlefastverb and handleslowverb in main.c.

BarChart uses the Applet Toolkit to implement its system event handler. See bcfastiacmessage () in barchart.c for details.

DocServer 2.0 also uses fast event handlers.

Limits of System Event Handlers

- You can reference your globals.

IAC Tools maintains the A5 register for you. Anything declared globally is available to your code when running from a system event handler.

- You cannot depend on the window list.

The sending app's window list is current. We've tried experiments to see if we can write into our own windows if we've saved WindowPtrs in A5-based globals. So far, no success.

- You cannot do any drawing.

This is completely uncharted territory. For now we are not doing any drawing from system event handlers.

- You cannot do anything with resources.

The resource map isn't yours. There may be some way to save and restore it, but we feel comfortable with this restriction. Resource-based operations can safely be done in scripts without calling any system event handlers.

- You can allocate, deallocate and reference things in your application heap.

We also maintain information about your heap zone, and save and restore the current heap zone. Without doing this, any allocation would be stored in the other app's heap.

Why system events are faster

Instead of being attached to your application, system event handlers belong to the operating system. They're much faster because there is no context switch. When the script running in Frontier calls one of your system event handlers, none of the heavy machinery of the Macintosh Process Manager gets invoked.

IACremovesystemhandlers ()

All apps that install system event handlers must call IACremovesystemhandlers before exiting. If they don't, system event handlers will accumulate in the system event table. If somehow one of these handlers were invoked, the system would almost certainly crash.

IACremovesystemhandlers is called by Trig at the end of main () in main.c.

IACremovesystemhandlers is called by the Applet Toolkit at the end of runapplet in appletmain.c.

Calling System Event Handlers from Frontier

From Frontier 2.0, you can invoke a system event handler by calling the new `systemEvent` verb.

Here's an example, the script that calls the `Trig` verb that determines the absolute value of a number:

```
on abs (x)
    return (systemEvent ('trig', 'abs ', '——', double (x)))
```

`Trig` installs two classes of handlers, one to handle its slower events and one to handle the faster events. The slower event handler's class is `TRIG` and the faster one's class is `trig`. We're also following that convention in the Applet Toolkit, all uppercase is the slower handler, all lowercase is the faster one.

Reorganized IAC Tools

In Frontier SDK 1.0, IAC Tools was implemented in a single C source file. In 2.0, there are 20 source files.

We reorganized for two reasons:

1. To take advantage of THINK C's "Smart Link" feature. If an application or code extension never calls routines in one of the files, the code for that file isn't linked into the resulting program or resource. This was especially important in implementing space-efficient UCMDs.
2. By having modular support for each data type (long, string, text, boolean, etc.) it makes it easier to upgrade IAC Tools to support additional data types. If you implement support for a new type, please consider sharing it with other users of the IAC Tools library.

Here's a list of the 20 files that make up the IAC Tools library:

1. iac.c — main file.
2. iac.h — header file, include this in all files that call iac.c routines.
3. iacapps.c -- routines that are likely to be called by a double-clickable application.
4. iacasynch.c — support for asynchronous Apple Events.
5. iacbinary.c — support for binary parameters and returned values.
6. iacboolean.c — support for boolean parameters and returned values.
7. iacdoubles.c — support for double parameters and returned values.
8. iachandler.c -- routines that install and remove Apple Event handlers.
9. iacinternal.h — function prototypes for routines in iacops.c.
10. iaclong.c — support for long parameters and returned values.
11. iacnewsystemverb.c -- the single system-handler call that UCMDs are likely to make.
Putting this routine in a separate file helps keep code extensions smaller.
12. iacops.c — basic support routines for all iacxxx.c files.
13. iacpoint.c — support for Point parameters and returned values.
14. iacreceive.c -- routines for code that receives Apple Events.
15. iacrect.c — support for Rect parameters and returned values.
16. iacsend.c -- routines for code that sends Apple Events.
17. iacshort.c — support for short parameters and returned values.
18. iacstring.c — support for string parameters and returned values.
19. iacsysevents.c — support for system event handlers.
20. iactext.c — support for text parameters and returned values.

Change Notes

Changed 'doub' to 'exte'

In Trig, double parameters were coming in from Frontier with the type 'exte' not 'doub.' In the Quick Script window I typed “doubleType” and got the value 'exte'. This nailed it for me. Could it be that the double routines in iac.c never worked?

Routines that work with double-precision floating point numbers now deal with parameters whose type is 'exte'. THINK C developers: be sure that Native Floating Point Format is checked in the Options dialog.

New field in IACglobals — idprocess

It's set by IACinit by asking the Process Manager for the id of the process we're running inside of. Menu Sharing Toolkit now uses this field to set the value of MSglobals.clientid.

Asynchronous events

Added support for asynchronous events in iacasynch.c. You create the message using IACnewverb or IACnewsystemverb, add parameters, and then call IACsendasynch to send the message. You provide a callback routine that gains control when the answer is received.

All messages sent "normal" priority

In IACnewverb, we used to send them with the constant kAEHighPriority, now we use the constant kAENormalPriority.

New iac.c routine IAChandlerinstalled

Makes it possible to find out if an event handler has been installed for a given class and id. This call is used by menusharing.c to determine if fast menu sharing is available.

New field in IACglobals — errorcode

Most iac.c routines just return true or false indicating whether they worked. In some cases you may need to know specifically why an operation failed. The error code from all iac.c routines is copied into IACglobals.errorcode for this purpose.

Changes from Leonard Rosenthol, Aladdin Systems

Leonard made several changes to IAC Tools and asked that they be integrated with the released version of iac.c/iac.h. Here's a list of the changes:

1. Streamlined IACreturnstring.
2. Added IACnextparamisoptional to mimic functionality that was available in UserLand IAC Toolkit.

Call this function before doing one of the IACgetxxxparam routines. If the param isn't there, the Apple Event handler will not return an error to the caller. Added a new global, IACglobals.nextparamoptional to support this.

Deleted IACpushCStringparam, IACgetCstringparam

They were there to support some early experimentation we were doing with Microsoft apps. The experiment didn't lead anywhere, so we axed'm.

IACreturnerror

In iac.c, if errs is nil or empty, then don't push the string onto the reply. The net: Frontier will interpret the error code as a Macintosh operating system error and display a more informative error message.

BarChart

Now implements a pair of system event handlers that allow a script to get the value of a bar very quickly, and to set the value of a bar.