Mindcraft, Inc.
170 Knowles Drive, Suite 206
Los Gatos, CA 95032-1833
(408) 364-2860 Fax: (408) 364-2862

Detailed QuickTime Plug-In Problems, Analyses, and Solutions

November 4, 1998

Contents

Section 1

Executive Summary

Microsoft hired Mindcraft, Inc. as an independent test lab with the following assignment:

- To find the causes of the problems Apple Computer, Inc. reported with Internet Explorer using Apple's QuickTime Web browser plug-in and
- To devise a solution that would make the QuickTime plug-in work properly with Internet Explorer, if such a solution exists.

This detailed report presents our findings in a way that technical people can understand. A less technical summary of our work is in a separate document entitled *Summary QuickTime Plug-In Problems, Analyses, and Solutions*.

Microsoft provided Mindcraft with a description of the problems Apple reported and with Microsoft's own findings. We told Microsoft that we would take a different approach to analyze and to solve the reported problems so that our assessment would be independent of theirs. If we could not take a different approach or come up with a different solution for any problem, we would carefully verify that Microsoft's assessment of the problem was correct and that their solution worked.

We decided early in our work to create our own plug-in so that we could study the interactions between the QuickTime plug-in and Internet Explorer. In addition, it would let us determine if the problem is specific to the QuickTime plug-in itself or something external to it. Just as importantly, it would let us see if Microsoft did something malicious to interfere with the QuickTime plug-in.

We tested whether or not Internet Explorer uses the QuickTime plug-in when it encounters an EMBED HTML tag. Our tests used Microsoft-supplied media files with the 11 filename extensions that the QuickTime plug-in supports. We found three test failures where Internet Explorer did not invoke the QuickTime Plug-in as expected. Each failure was the result of one or two Apple errors as shown in Table 1. Apple had the knowledge to correct these errors because, as Table 1 illustrates, Apple properly set the information to tell Internet Explorer to use the QuickTime plug-in for 8 of the 11 test files.

• Table 1: Apple's QuickTime Plug-In Errors
(Apple's errors are indicated by a **No** in the table)

Did Apple Do It?	Filename Extensions QuickTime Plug-In Supports										
	QT	VFW	AIFC	AIF	AIFF	AU	AVI	FLC	MID	MOV	WAV
Set the FileExtents resource in the QuickTime Plug-in for each filename extension it is to support, as the Netscape Plug-in specification requires	**No**	**No**	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Set the EnablePlugin Registry key for each filename extension the QuickTime Plug-in is to support, as ActiveX controls require	**Yes**	**No**	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Test Results											
Before Mindcraft's fix	**Fail**	**Fail**	**Fail**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**
After Mindcraft's fix to repair FileExtents and EnablePlugin errors	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**	**Pass**

Based on our detailed analysis of the problems, we conclude that Microsoft did nothing malicious to cause the QuickTime plug-in to fail. In addition, we find that Apple has bugs in its QuickTime plug-in and its installer that prevent the plug-in from working properly.

Section
2

Problem Analysis

The goal of our problem analysis is to find the root causes of the problems Apple reported with the QuickTime plug-in when it is used with Internet Explorer. Once we understand these causes then we can devise a fix for the problems, if possible.

Microsoft provided Mindcraft with 11 test cases for the EMBED HTML tag that they said would reproduce the problems that Apple reported. We will call these test cases the QuickTime/Internet Explorer test list, or QT/IE test list for short. *The QuickTime/Internet Explorer Test List* section below gives the details about the test cases.

Background

We believe that it is important to give you some background information about the essential elements involved with the problems and the solutions. Therefore, we will describe what plug-ins, ActiveX controls, MIME types, file extensions, and the Registry are.

Plug-Ins

A plug-in is a program that gives a Web browser capabilities it did not have initially. Adding a plug-in to a browser is like adding a compact disk (CD) player to your home stereo system. You put the CD player in the cabinet with the rest of your stereo system and then connect it properly to your system. Then when you select the CD player to be the source of your music and put a CD in it, you will hear the music coming from the CD. The QuickTime plug-in works just like a CD player. After it is installed correctly into Internet Explorer, you can play music, movies, and other multimedia files via the QuickTime plug-in. You should note plug-ins that meet the Netscape specification work only in a Web browser.

Netscape Communications Corporation created the concept of plug-ins for its Navigator browser. Microsoft put the capability to use plug-ins in Internet Explorer 3.0 and 4.0. Microsoft followed Netscape's plug-in specification. In theory, this should allow Internet Explorer 3.0 through the latest release of Internet Explorer 4 to run the same plug-ins as Netscape Navigator does.

The latest version of the QuickTime plug-in is 2.0.1. It comes as part of the most current release of Apple's QuickTime 3.

ActiveX Controls

ActiveX controls are another way to extend the capabilities of a Web browser. Using the CD player analogy again, adding an ActiveX control to a Web browser is like adding a portable CD player to your home stereo system. You connect the portable CD player to the stereo system, set the music source to be the CD player, insert a CD, and enjoy the music. Similarly, an ActiveX control works much like a portable CD player. After it is installed correctly into Internet Explorer, it gives Internet Explorer a new capability as if it

were originally built-in. The significant difference between a plug-in and an ActiveX control is that an ActiveX controls can used by other programs in addition to Web browsers.

Filename Extensions

Files on hard disks, floppy disks, and other devices have names that consist of two parts, the filename and the filename extension. For example, a file named “report.doc” has “doc” as its filename extension. A filename extension is the set of characters that follow the last period in the name of a file.

Filename extensions are important because they can tell Windows the type of file and the application used to create the file. In our example, Microsoft Word uses the “doc” filename extension to mark files it creates. This association between filename extension and application is important. For example, when you double click on a file’s icon, Windows will look up the filename extension in a table and automatically start the application associated with that type of file.

The association between filename extension and the application that uses files of that type applies to Internet Explorer and other Web browsers. Internet Explorer uses the filename extension and MIME type (which is defined in the next section) to determine which built-in capability, plug-in, or ActiveX control to call when it encounters a file with a specific filename extension.

MIME Types

MIME is an acronym for Multipurpose Internet Mail Extension. It is the standard way a Web server tells a Web browser what kind of file it is sending and is used similarly by other applications. MIME is an extension of the Internet mail protocol that enables sending e-mail messages or files containing extended character sets (which are useful for foreign languages), voice mail, facsimile images, music, movies, and so on.

A MIME type specifies a generic type of information in a file. It does not tell a browser what application it should use to read or display the file. A Web browser must look at a file’s MIME-type and the filename extension as well as its own configuration to determine whether it should display the file using a built-in capability, a plug-in, or an ActiveX control.

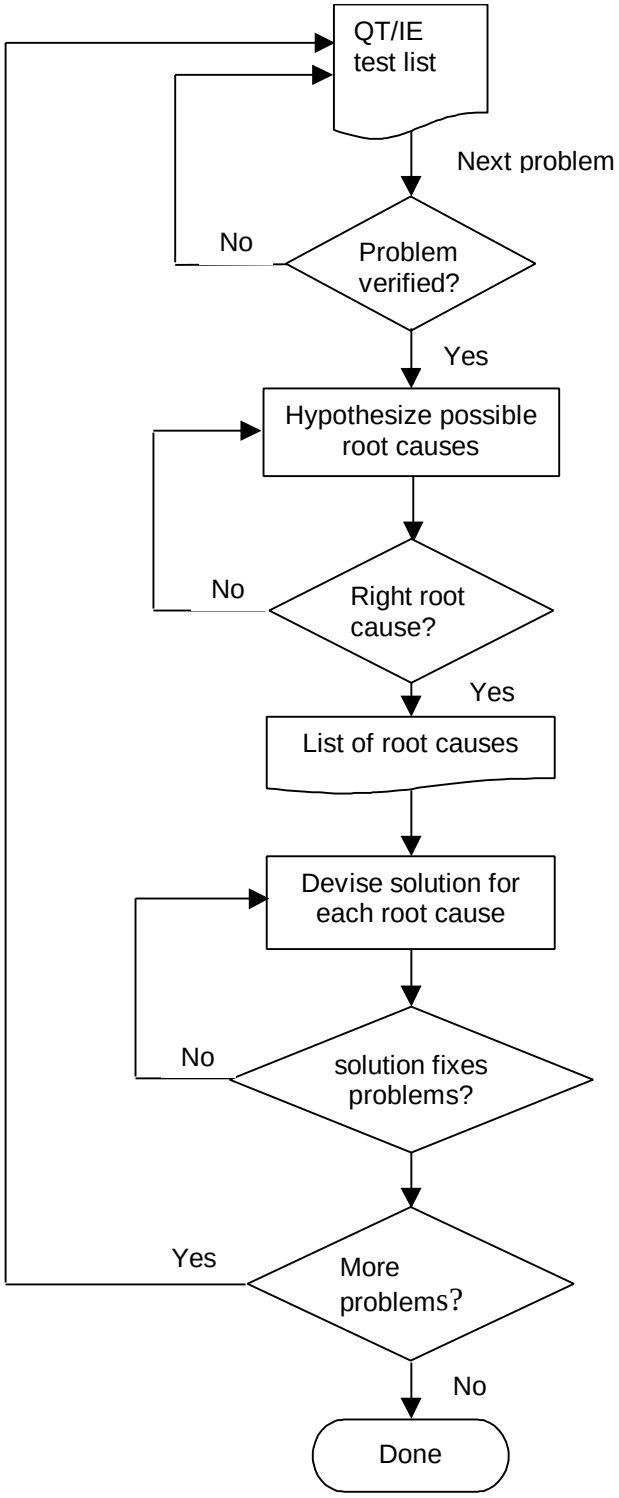
Registry

The Windows family of operating systems provides a built-in database where applications can store configuration parameters. This database is simply called the Registry. It is where Internet Explorer stores its associations of MIME types and filename extensions that determine which built-in, plug-in or ActiveX control it will use.

Problem Analysis Strategy

Figure 1 shows the process we used to analyze the QuickTime plug-in problems Apple reported.

• Figure 1: Problem Analysis Process



Possible Causes of the Problems

The problems Apple reported could be caused by one of the following:

- The Windows operating system;
- Internet Explorer;
- The QuickTime plug-in;
- The QuickTime plug-in installer; or
- A combination of the preceding.

The following sections will show how each of the above could be the source of a reported problem.

How the Windows Operating System Could Cause a Problem

The only way that a Windows operating system could cause the problems observed when the QuickTime plug-in is used with Internet Explorer would be if Microsoft intentionally detected that a non-Microsoft plug-in was being installed or called and used ActiveMovie instead.

In order to detect such malicious deeds, we decided to build our own QuickTime plug-in installation program and to use it to install Apple's QuickTime plug-in correctly. If Microsoft were being malicious, it would have to allow some flexibility in detecting when a QuickTime plug-in was being installed and used in order to prevent future versions from working properly. Therefore, if our installation program was correct and Microsoft did not do anything malicious, the real QuickTime plug-in we install should work properly. If we find a problem with the installation, we will know that Microsoft somehow interfered with our program. If the installation is correct but the QuickTime plug-in does not work, we can verify if Microsoft is interfering with the plug-in by creating our own plug-in that will print debugging information. We would use our own QuickTime installation program to install it.

Another thing we did to detect malicious deeds was to build our pseudo-QuickTime plug-in. If the OS or web browser thought it was QuickTime and treated it maliciously then that would have told us something. You may wonder why we went to the trouble to make our own QuickTime plug-in installation program. Why didn't we use the real thing? Simply put, we did not have access to the source code. Our only option was to simulate the QuickTime plug-in installation program as closely as possible while adding any corrections needed based on our testing.

How Internet Explorer Could Cause a Problem

The most obvious ways that Internet Explorer could be causing problems with the QuickTime plug-in are:

1) It could be misreading the table in the Registry that contains the associations between MIME types and filename extensions because of a bug (this could result in calling something other than the QuickTime plug-in);

2) It could read the MIME type/filename extension table in the Registry correctly but not call the QuickTime plug-in properly because of a bug;

3) It could see in the MIME type/filename extension table in the Registry that the QuickTime plug-in is to be called but maliciously call Microsoft's ActiveMovie instead; or

4) It could see a problem in the MIME type/filename extension table in the Registry or a problem with the QuickTime plug-in and do something it determines to be safer for the user.

To be able to detect each of the above possibilities, we used the following tools and procedures:

- **Our pseudo-QuickTime plug-in**
  This lets us see if Internet Explorer is calling the plug-in and the MIME type and filename extension with which it is being called. This tool will let us see if any of the above possibilities are happening. Our pseudo-QuickTime plug-in is based on the "WinTemplate" sample provided with the Netscape Plug-in Software Development Kit (available from ftp://ftp.netscape.com/pub/sdk/plug-in/windows/oct_21_97/winNPSDK.zip). Netscape provides the "WinTemplate" example as a starting point for developing plug-ins. Our pseudo-QuickTime plug-in has the capability to:

  1) Print out the MIME type that the web server passed to the plug-in.

  2) Print out whether the plug-in was invoked through an EMBED HTML tag.

  3) Print out the name of the file that Internet Explorer passed to the plug-in to play.

  Our simple plug-in did nothing more than report this data.

- **RegDump**
  This program dumps the registry and compares two dumps to show the differences. We use this tool to dump the Registry right after we install each version of Internet Explorer we will test. Then we will dump the registry after we install the QuickTime plug-in. When we compare the two dumps, we will be able to see what the QuickTime installer changed. This will help us determine if the QuickTime installer is setting the Registry properly.

- **RegMon**
  This program monitors in real-time any changes made to the Registry and logs any Registry accesses. RegMon lets us see if Internet Explorer modifies the QuickTime installer Registry changes.

- **DumpBin**
  This program allows us to see the contents of a program as ASCII characters (but it is not a disassembler). We can use it to see if the QuickTime plug-in program contains items required by the Netscape plug-in specification.

- **RegEdit**
  We used the standard Microsoft RegEdit utility to examine the Registry and to perform experiments such as adding missing Registry keys.

How the QuickTime Plug-In Could Cause a Problem

The QuickTime plug-in could cause a problem if it:

1) Has a bug that causes it fail;

2) Has a bug that causes it fail and Internet Explorer uses ActiveMovie instead; or

3) Does not correctly follow the Netscape plug-in specification and Internet Explorer detects the problem.

To analyze whether any of the above is a possible cause of a problem, we would need to have access to the source code for Internet Explorer and the QuickTime plug-in. Because the source code is not available to us, our only option is to look at the resources in the QuickTime plug-in to see if those could be the source of a problem. While this is only a part of the Netscape plug-in specification, it is something that could account for some of the problems reported.

How the QuickTime Plug-In Installer Could Cause a Problem

The QuickTime plug-in installer could cause problems if it incorrectly sets up the MIME type and filename extension associations in the Registry. If these associations are incorrect, Internet Explorer may use its built-in capability, an ActiveX control, or another plug-in to handle a file with a MIME type/filename extension that Apple expected QuickTime to handle.

We will look at the Registry using the RegEdit, RegMon, and RegDump tools to see if it is configured consistent with Microsoft's requirements for Internet Explorer plug-ins.

How a Combination of the Above Could Cause a Problem

There are many ways that combinations of any of the above may cause any of the problems that Apple reported. Therefore, as we analyze the possible causes of each reported problem, we will be sure to look at multiple causes as the source of a problem.

Section
3

Verifying the Problems

Table 2 shows the combinations of operating systems and Web browsers we tested against the QT/IE test list.

- Table 2: Operating Systems and Web Browsers Tested

Operating System	Web Browser
Windows 95 (Version 4.00.950a)	Netscape Navigator 4.5 Internet Explorer 4.0 (4.72.3110.8)
Windows 98 (Version 4.10.1998)	Netscape Navigator 4.5 Internet Explorer 4.0 (4.72.3110)
Windows NT Server 4.0 (Version 4.0, Build 1381 with Service Pack 3)	Netscape Navigator 4.5 Internet Explorer 3.02 (4.70.1300) Internet Explorer 4.0 (4.72.3110.8)

We used the computer configuration shown in Table 3 for all of our tests.

- Table 3: Test System

System Component	Description
CPU	266 MHz Pentium® II; Intel® AL440LX motherboard
RAM	64 MB EDO RAM
Disk	4 GB ATA-33

The QuickTime/Internet Explorer Test List

Table 4 shows the QuickTime/Internet Explorer test list. It specifies the filename extensions to test with the QuickTime plug-in. We put the media files Microsoft provided us on an appropriately configured Web server along with a special Web page containing links to all the files on the list. We will verify each test case on the list by installing the QuickTime plug-in into the appropriate Web browser and pointing the browser to the special test Web page. The behavior when we click each link tells us if there is a problem.

• Table 4: The QuickTime/Internet Explorer Test List

Test List	
Filename Extension	Test With EMBED tag?
AVI (video)	Yes
VFW (video)	Yes
FLC (video)	Yes
MOV (video)	Yes
QT (video)	Yes
AIF (audio)	Yes
AIFF (audio)	Yes
AIFC (audio)	Yes
AU (audio)	Yes
MID (audio)	Yes
WAV (audio)	Yes

Test Results

Table 5 shows the test results for all of the operating system/Web browser combinations we tested. You should note that each version of Internet Explorer behaved the same on all operating systems. If you want to duplicate our tests, be sure to run the QuickTime control panel and put a check mark next to all of the file associations shown before running the tests.

• Table 5: QuickTime Plug-in Test Results

	Netscape Navigator 4.5	Microsoft Internet Explorer 3.x and 4.x
Filename Extension	OK With EMBED tag?	OK With EMBED tag?
AVI (video)	Yes	Yes
VFW (video)	Yes	Note 2
FLC (video)	Yes	Yes
MOV (video)	Yes	Yes
QT (video)	Yes	Note 2
AIF (audio)	Yes	Yes
AIFF (audio)	Yes	Yes
AIFC (audio)	Yes	Note 2
AU (audio)	Yes	Yes
MID (audio)	Note 1	Yes
WAV (audio)	Yes	Yes

Notes:

1) QuickTime plug-in not used by the web browser. Browser prompted to open file or save to disk. If open was selected, ActiveMovie was used.

2) Used ActiveMovie to play the file automatically instead of using the QuickTime plug-in.

Note 2 in Table 5 indicates a problem with how the QuickTime plug-in plays certain types of files in Internet Explorer. The Internet Explorer problems are that it:

1) Does not use the QuickTime plug-in to play QT files.

2) Does not use the QuickTime plug-in to play VFW files.

3) Does not use the QuickTime plug-in to play AIFC files.

The following sections will analyze each of the above problems.

QuickTime Plug-In Does Not Play QT Files

Internet Explorer will not use a plug-in to play files with a specific filename extension unless that plug-in indicates that it supports these files. Microsoft thought that the QuickTime plug-in did not specify its support for "qt" files and, thus, was not being called by Internet Explorer to play a file with a "qt" filename extension.

We started investigating this possibility by studying Netscape's specification for web browser plug-ins (http://developer.netscape.com/docs/manuals/communicator/plug-in/index.html). This specification states that the Windows version information contained in the plug-in dynamic link library (DLL) must list the filename extensions and MIME types that the plug-in can handle. Specifically, a Web browser will examine the FileExtents and MIMETypes strings in the Windows version information to learn what files the plug-in claims to support. According to the Netscape specification, if the QuickTime plug-in does not list "qt" among the filename extensions it supports then a Web browser may not call the plug-in for files that end with a "qt" filename extension.

To investigate this possibility, we used the DumpBin program to examine the contents of the QuickTime plug-in (npqtplug-in.dll). We found that the plug-in does not list "qt" in its FileExtents (see Appendix 1 for detailed output from DumpBin). To confirm that the QuickTime plug-in did not notify a Web browser that it could handle files with the "qt" extension, we used the special "About: plug-ins" URL supported by the Netscape Navigator. A screen shot of the output, in Appendix 2, shows that the QuickTime plug-in does not correctly tell a Web browser that it can handle "qt" files.

If a Web server sends a MIME type supported by the QuickTime plug-in along with a "qt" file then Netscape Navigator will use the QuickTime plug-in to play it. We verified that Netscape Navigator would not play a "qt" file stored on a local disk with the QuickTime plug-in because the file does not have a MIME type.

As a final verification that Internet Explorer was unaware that the QuickTime plug-in could play "qt" files and that the QuickTime plug-in itself was the source of the problem, we created our own plug-in that included "qt" in its FileExtents string (which tells a Web browser that the plug-in supports "qt" files). We ran the QT test after installing our plug-in into Internet Explorer. The result was that Internet Explorer used our plug-in to handle the "qt" file.

Our conclusion is that the missing "qt" in the QuickTime plug-in FileExtents string caused it to fail the QT test.

QuickTime Plug-In Does Not Play VFW Files

Our analysis of the problem with “qt” files revealed that a similar problem exists for “vfw” files. The QuickTime plug-in also omitted “vfw” from its FileExtents string (see Appendix 1).

Even if Apple had set “vfw” in the FileExtents string, the QuickTime plug-in still would not have been called because ActiveX controls take precedence over plug-ins for the same MIME type or filename extension. Setting the EnablePlugin Registry key for the ActiveX control will disable the ActiveX precedence. Despite setting the EnablePlugin Registry key for other file types, the QuickTime plug-in installer did not set it for the “vfw” filename extension. We determined this using RegDump to compare the Registry before and after the QuickTime plug-in installation. Appendix 5 shows the EnablePlugin settings that the QuickTime plug-in installer set as logged by RegMon.

These are the reasons why Internet Explorer did not use the QuickTime plug-in to play the “vfw” file.

QuickTime Plug-In Does Not Play AIFC Files

As with “qt” and “vfw” files, the QuickTime plug-in omitted “aifc” from its FileExtents string (see Appendix 1). In addition, the QuickTime plug-in installer did not set the EnablePlugin Registry key for the “aifc” filename extension (see Appendix 5).

These are the reasons why Internet Explorer did not use the QuickTime plug-in to play the “aifc” file.

Section

4

Solutions to the Problems

Registry Fixes

Missing QT, VFW, and AIFC in FileExtents

Because we did not have access to Apple's source code for its QuickTime plug-in, we could not correct the FileExtents string of the version information to add the missing "qt", "vfw", and ".aifc" file extensions. We learned from Microsoft engineers that when Internet Explorer starts up, it creates its own set of registry keys for file extensions and MIME types supported by its plug-ins. It does this under the Registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Plugins\Extension\<extension>

Internet Explorer creates the following values under the above Registry key:

[default] = "QuickTime Plug-In"

Content Type = <mimetype>

Version="2.0.1"

Location=<location to QuickTime DLL >

We can fix the deficiencies in the QuickTime plug-in by creating registry entries for each missing filename extension. However, Microsoft engineers told us that Internet Explorer would remove Registry keys for file extensions not listed in the FileExtents of the installed plug-ins. Therefore, the solution is to stop Internet Explorer from removing these additions. This can be accomplished by setting the following Registry key with the default value shown:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\nternet Explorer\Plugins\DisableRegistryInitOnStartup = "true"

This registry key is not documented. However, a correct plug-in will not need to set it because the plug-in will specify all of its supported filename extensions in the FileExtents string of the plug-in version information.

ActiveX Controls Take Precedence

It is undocumented that Internet Explorer gives an ActiveX control precedence over a plug-in when both handle the same file extension or MIME type (below we will show you that Apple knows how to fix this behavior). If you want to disable ActiveX precedence for a given filename extension and MIME type, modify the Registry as follows:

1) Under the "HKEY_CLASSES_ROOT\MIME\Database\Content Type" key, look for a key that matches the MIME type for which you want to disable ActiveX precedence. Find the "CLSID" value for that key.

2) Go to the “HKEY_CLASSES_ROOT\<CLSID>\” key and find the CLSID from step #1.

3) For that CLSID create an “EnablePlugin” key, if it does not already exist.

4) Under the “EnablePlugin” key from step #3, create a new key for the filename extension you want the plug-in to handle. This new key should not have any default value.

5) Under the “EnablePlugin” key from step #3, create a “MIME” key, if it does not already exist.

6) Finally, under the MIME key from step #5, create a new key with the name of the MIME type for which you want to disable ActiveX precedence. This new key should not have any default value.

For example, if you want to disable ActiveX precedence for the filename extension ".aifc" and its associated MIME type "audio/aiff" then we would need to perform the following steps in the registry:

1) Go to the key "HKEY_CLASSES_ROOT\MIME\Database\Content Type" and find that the CLSID value is set to "{05589fa1-c356-11ce-bf01-00aa0055595a}".

2) Go to the key "HKEY_CLASSES_ROOT\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}"

3) If the EnablePlugin key does not already exist, then create it.

4) Create the key "HKEY_CLASSES_ROOT\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.aifc".

5) Create the key: "HKEY_CLASSES_ROOT\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\MIME"

   if it does not already exist.

6) Create the key "HKEY_CLASSES_ROOT\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\MIME\audio/aiff"

After you make the above Registry changes, the plug-in will take precedence over an ActiveX control that is registered to handle the same filename extension or MIME type. The above EnablePlugin Registry key is undocumented. Apple knew about this Registry key.

We used RegDump tool to compare the contents of the Registry before and after the QuickTime plug-in was installed. Additionally, the RegMon tool provided us with a real-time view of all Registry accesses that occurred during QuickTime installation. We could see that the QuickTime installer set the EnablePlugin Registry key for 9 of 11 filename extensions but not for the “aifc” and “vfw” filename extensions. (See Appendix 5 for a list of EnablePlugin Registry changes that the QuickTime installer made).

Mindcraft's QuickTime Plug-In Installer

In order to validate our solutions, we made an installer for the QuickTime plug-in that uses all of our above Registry fixes. This InstallShield program installs the QuickTime plug-in into both Internet Explorer and Netscape Navigator. Our installer demonstrates that:

1. It is possible to install a plug-in so that it works as expected, and

2. The Windows operating system and Internet Explorer do nothing malicious to prevent the QuickTime plug-in from installing and working correctly.

Please note that this is not a complete installation for QuickTime but rather just a demonstration of how to install a plug-in.

Mindcraft's QuickTime Plug-In Fix

We provide an InstallShield5 program in Appendix 4 that installs all of the fixes outlined in this report. This installer is available at our web site (http://www.mindcraft.com/qtfix). It must be run after QuickTime has been installed on your computer and you must run the QuickTime control panel and put a check mark next to all of the file type associations shown before running the tests.

Table 6 shows the results of running through the QT/IE test list after installing our fix.

- Table 6: Post Fix Verification Test Results

Filename Extension	Netscape Navigator 4.5 OK With EMBED tag?	Microsoft Internet Explorer 3.x and 4.x OK With EMBED tag?
AVI (video)	Yes	Yes
VFW (video)	Yes	Yes
FLC (video)	Yes	Yes
MOV (video)	Yes	Yes
QT (video)	Yes	Yes
AIF (audio)	Yes	Yes
AIFF (audio)	Yes	Yes
AIFC (audio)	Yes	Yes
AU (audio)	Yes	Yes
MID (audio)	Yes	Yes
WAV (audio)	Yes	Yes

Section

5

About Mindcraft

Mindcraft® has been providing software testing and development services since 1985. With our focus on Internet and intranet technologies, we offer our vendor and user clients cost-effective services and products using the latest technology.

Mindcraft's services and products include:

- Performance testing for Web servers, LDAP directory servers, proxy servers, and e-mail servers.
- Benchmark tools including WebStone, DirectoryMark, and others.
- Capacity planning
- Consulting
- Application testing
- Custom test development

Mindcraft is the only test lab to be a member of the Standard Performance Evaluation Corporation (SPEC). One of our employees also is a member of SPEC's Board of Directors.

Mindcraft has grown to become the largest Accredited POSIX Testing Laboratory in the world. As part of our accreditation, we developed a rigorous quality system that meets international standards. Our quality system is the cornerstone of all of the work we do.

If you want more information about this report, please contact Bruce Weiner by phone at (408) 364-2860 or by email at bruce@mindcraft.com.

Appendix

1

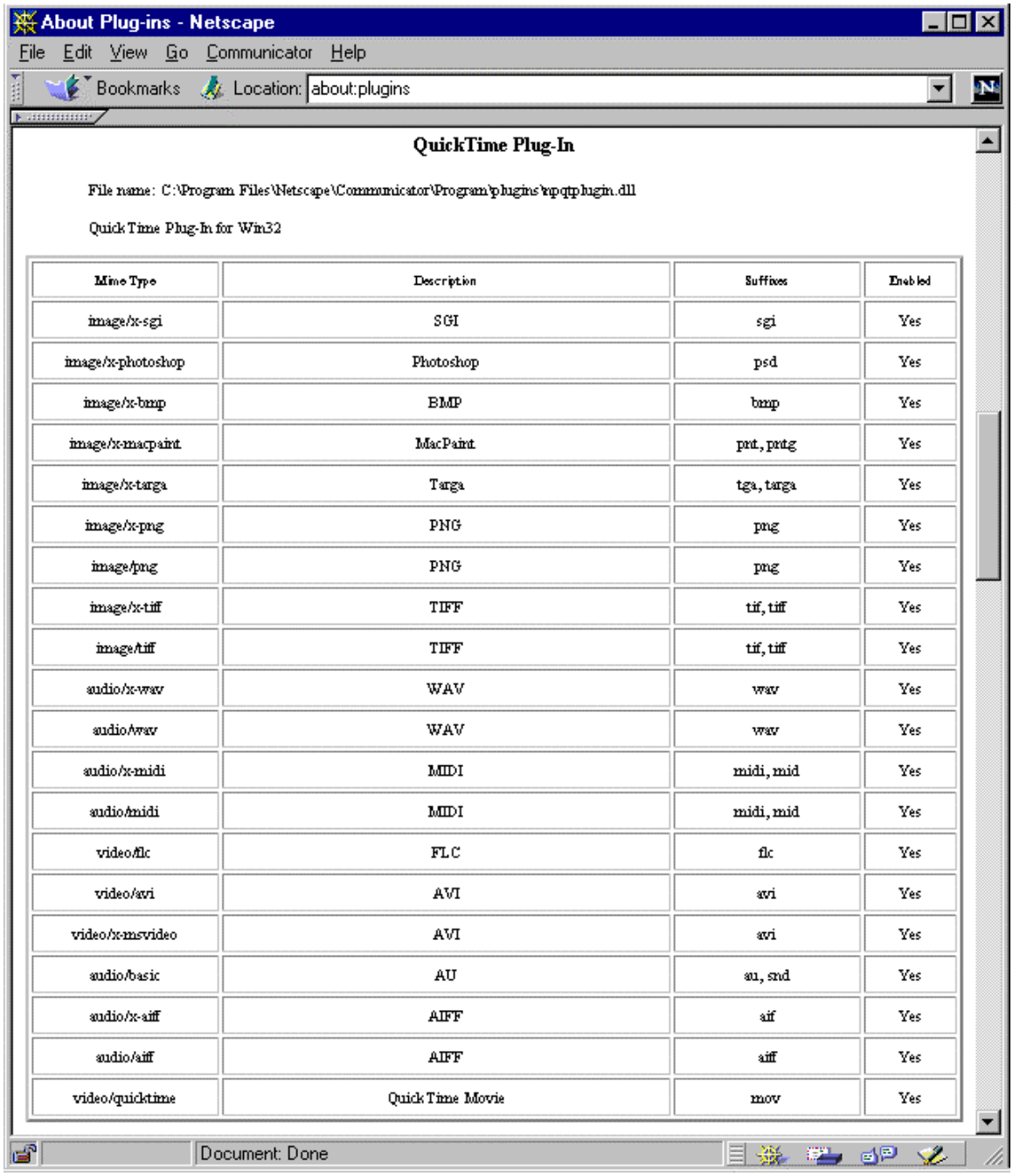
Appendix 1: QuickTime Plug-In Resources

```
100212D0: 32 00 00 00 06 01 73 00 01 00 46 00 69 00 6C 00  2.....s...F.i.l.
100212E0: 65 00 45 00 78 00 74 00 65 00 6E 00 74 00 73 00  e.E.x.t.e.n.t.s.
100212F0: 00 00 00 00 6D 00 6F 00 76 00 7C 00 61 00 69 00  ....m.o.v.|.a.i.
10021300: 66 00 66 00 7C 00 61 00 69 00 66 00 7C 00 61 00  f.f.|.a.i.f.|.a.
10021310: 75 00 2C 00 73 00 6E 00 64 00 7C 00 61 00 76 00  u.,.s.n.d.|.a.v.
10021320: 69 00 7C 00 61 00 76 00 69 00 7C 00 66 00 6C 00  i.|.a.v.i.|.f.l.
10021330: 63 00 7C 00 6D 00 69 00 64 00 69 00 2C 00 6D 00  c.|.m.i.d.i.,.m.
10021340: 69 00 64 00 7C 00 6D 00 69 00 64 00 69 00 2C 00  i.d.|.m.i.d.i.,.
10021350: 6D 00 69 00 64 00 7C 00 77 00 61 00 76 00 7C 00  m.i.d.|.w.a.v.|.
10021360: 77 00 61 00 76 00 7C 00 74 00 69 00 66 00 2C 00  w.a.v.|.t.i.f.,.
10021370: 74 00 69 00 66 00 66 00 7C 00 74 00 69 00 66 00  t.i.f.f.|.t.i.f.
10021380: 2C 00 74 00 69 00 66 00 66 00 7C 00 70 00 6E 00  ,.t.i.f.f.|.p.n.
10021390: 67 00 7C 00 70 00 6E 00 67 00 7C 00 74 00 67 00  g.|.p.n.g.|.t.g.
100213A0: 61 00 2C 00 74 00 61 00 72 00 67 00 61 00 7C 00  a.,.t.a.r.g.a.|.
100213B0: 70 00 6E 00 74 00 2C 00 70 00 6E 00 74 00 67 00  p.n.t.,.p.n.t.g.
100213C0: 7C 00 62 00 6D 00 70 00 7C 00 70 00 73 00 64 00  |.b.m.p.|.p.s.d.
100213D0: 7C 00 73 00 67 00 69 00 00 00 00 00 1E 02 FF 00  |.s.g.i.......ÿ.
100213E0: 01 00 46 00 69 00 6C 00 65 00 4F 00 70 00 65 00  ..F.i.l.e.O.p.e.
100213F0: 6E 00 4E 00 61 00 6D 00 65 00 00 00 51 00 75 00  n.N.a.m.e...Q.u.
10021400: 69 00 63 00 6B 00 54 00 69 00 6D 00 65 00 20 00  i.c.k.T.i.m.e. .
10021410: 4D 00 6F 00 76 00 69 00 65 00 28 00 2A 00 2E 00  M.o.v.i.e.(.*...
10021420: 6D 00 6F 00 76 00 29 00 7C 00 41 00 49 00 46 00  m.o.v.).|.A.I.F.
10021430: 46 00 28 00 2A 00 2E 00 61 00 69 00 66 00 29 00  F.(.*...a.i.f.).
10021440: 7C 00 41 00 49 00 46 00 46 00 28 00 2A 00 2E 00  |.A.I.F.F.(.*...
10021450: 61 00 69 00 66 00 29 00 7C 00 41 00 55 00 28 00  a.i.f.).|.A.U.(.
10021460: 2A 00 2E 00 61 00 75 00 2C 00 2A 00 2E 00 73 00  *...a.u.,.*...s.
10021470: 6E 00 64 00 29 00 7C 00 41 00 56 00 49 00 28 00  n.d.).|.A.V.I.(.
10021480: 2A 00 2E 00 61 00 76 00 69 00 29 00 7C 00 41 00  *...a.v.i.).|.A.
10021490: 56 00 49 00 28 00 2A 00 2E 00 61 00 76 00 69 00  V.I.(.*...a.v.i.
100214A0: 29 00 7C 00 46 00 4C 00 43 00 28 00 2A 00 2E 00  ).|.F.L.C.(.*...
100214B0: 66 00 6C 00 63 00 29 00 7C 00 4D 00 49 00 44 00  f.l.c.).|.M.I.D.
100214C0: 49 00 28 00 2A 00 2E 00 6D 00 69 00 64 00 29 00  I.(.*...m.i.d.).
100214D0: 7C 00 4D 00 49 00 44 00 49 00 28 00 2A 00 2E 00  |.M.I.D.I.(.*...
100214E0: 6D 00 69 00 64 00 29 00 7C 00 57 00 41 00 56 00  m.i.d.).|.W.A.V.
100214F0: 28 00 2A 00 2E 00 77 00 61 00 76 00 29 00 7C 00  (.*...w.a.v.).|.
10021500: 57 00 41 00 56 00 28 00 2A 00 2E 00 77 00 61 00  W.A.V.(.*...w.a.
10021510: 76 00 29 00 7C 00 54 00 49 00 46 00 46 00 28 00  v.).|.T.I.F.F.(.
10021520: 2A 00 2E 00 74 00 69 00 66 00 29 00 7C 00 54 00  *...t.i.f.).|.T.
10021530: 49 00 46 00 46 00 28 00 2A 00 2E 00 74 00 69 00  I.F.F.(.*...t.i.
10021540: 66 00 29 00 7C 00 50 00 4E 00 47 00 28 00 2A 00  f.).|.P.N.G.(.*.
10021550: 2E 00 70 00 6E 00 67 00 29 00 7C 00 50 00 4E 00  ..p.n.g.).|.P.N.
10021560: 47 00 28 00 2A 00 2E 00 70 00 6E 00 67 00 29 00  G.(.*...p.n.g.).
10021570: 7C 00 54 00 61 00 72 00 67 00 61 00 28 00 2A 00  |.T.a.r.g.a.(.*.
10021580: 2E 00 74 00 67 00 61 00 29 00 7C 00 4D 00 61 00  ..t.g.a.).|.M.a.
10021590: 63 00 50 00 61 00 69 00 6E 00 74 00 28 00 2A 00  c.P.a.i.n.t.(.*.
100215A0: 2E 00 70 00 6E 00 74 00 29 00 7C 00 42 00 4D 00  ..p.n.t.).|.B.M.
100215B0: 50 00 28 00 2A 00 2E 00 62 00 6D 00 70 00 29 00  P.(.*...b.m.p.).
100215C0: 7C 00 50 00 68 00 6F 00 74 00 6F 00 73 00 68 00  |.P.h.o.t.o.s.h.
```

```
100215D0: 6F 00 70 00 28 00 2A 00 2E 00 70 00 73 00 64 00  o.p.(.*...p.s.d.
100215E0: 29 00 7C 00 53 00 47 00 49 00 28 00 2A 00 2E 00  ).|.S.G.I.(.*...
100215F0: 73 00 67 00 69 00 29 00 00 00 00 00 2C 00 06 00  s.g.i.).....,...
10021600: 01 00 46 00 69 00 6C 00 65 00 56 00 65 00 72 00  ..F.i.l.e.V.e.r.
10021610: 73 00 69 00 6F 00 6E 00 00 00 00 00 32 00 2E 00  s.i.o.n.....2...
10021620: 30 00 2E 00 31 00 00 00 44 00 12 00 01 00 49 00  0...1...D.....I.
10021630: 6E 00 74 00 65 00 72 00 6E 00 61 00 6C 00 4E 00  n.t.e.r.n.a.l.N.
10021640: 61 00 6D 00 65 00 00 00 51 00 75 00 69 00 63 00  a.m.e...Q.u.i.c.
10021650: 6B 00 54 00 69 00 6D 00 65 00 20 00 50 00 6C 00  k.T.i.m.e. .P.l.
10021660: 75 00 67 00 2D 00 49 00 6E 00 00 00 66 00 21 00  u.g.-.I.n...f.!.
10021670: 01 00 4C 00 65 00 67 00 61 00 6C 00 43 00 6F 00  ..L.e.g.a.l.C.o.
10021680: 70 00 79 00 72 00 69 00 67 00 68 00 74 00 00 00  p.y.r.i.g.h.t...
10021690: A9 00 20 00 41 00 70 00 70 00 6C 00 65 00 20 00  ©. .A.p.p.l.e. .
100216A0: 43 00 6F 00 6D 00 70 00 75 00 74 00 65 00 72 00  C.o.m.p.u.t.e.r.
100216B0: 2C 00 20 00 49 00 6E 00 63 00 2E 00 20 00 31 00  ,. .I.n.c... .1.
100216C0: 39 00 39 00 32 00 2D 00 31 00 39 00 39 00 38 00  9.9.2.-.1.9.9.8.
100216D0: 00 00 00 00 12 02 FD 00 01 00 4D 00 49 00 4D 00  ......ý...M.I.M.
100216E0: 45 00 54 00 79 00 70 00 65 00 00 00 76 00 69 00  E.T.y.p.e...v.i.
100216F0: 64 00 65 00 6F 00 2F 00 71 00 75 00 69 00 63 00  d.e.o./.q.u.i.c.
10021700: 6B 00 74 00 69 00 6D 00 65 00 7C 00 61 00 75 00  k.t.i.m.e.|.a.u.
10021710: 64 00 69 00 6F 00 2F 00 61 00 69 00 66 00 66 00  d.i.o./.a.i.f.f.
10021720: 7C 00 61 00 75 00 64 00 69 00 6F 00 2F 00 78 00  |.a.u.d.i.o./.x.
10021730: 2D 00 61 00 69 00 66 00 66 00 7C 00 61 00 75 00  -.a.i.f.f.|.a.u.
10021740: 64 00 69 00 6F 00 2F 00 62 00 61 00 73 00 69 00  d.i.o./.b.a.s.i.
10021750: 63 00 7C 00 76 00 69 00 64 00 65 00 6F 00 2F 00  c.|.v.i.d.e.o./.
10021760: 78 00 2D 00 6D 00 73 00 76 00 69 00 64 00 65 00  x.-.m.s.v.i.d.e.
10021770: 6F 00 7C 00 76 00 69 00 64 00 65 00 6F 00 2F 00  o.|.v.i.d.e.o./.
10021780: 61 00 76 00 69 00 7C 00 76 00 69 00 64 00 65 00  a.v.i.|.v.i.d.e.
10021790: 6F 00 2F 00 66 00 6C 00 63 00 7C 00 61 00 75 00  o./.f.l.c.|.a.u.
100217A0: 64 00 69 00 6F 00 2F 00 6D 00 69 00 64 00 69 00  d.i.o./.m.i.d.i.
100217B0: 7C 00 61 00 75 00 64 00 69 00 6F 00 2F 00 78 00  |.a.u.d.i.o./.x.
100217C0: 2D 00 6D 00 69 00 64 00 69 00 7C 00 61 00 75 00  -.m.i.d.i.|.a.u.
100217D0: 64 00 69 00 6F 00 2F 00 77 00 61 00 76 00 7C 00  d.i.o./.w.a.v.|.
100217E0: 61 00 75 00 64 00 69 00 6F 00 2F 00 78 00 2D 00  a.u.d.i.o./.x.-.
100217F0: 77 00 61 00 76 00 7C 00 69 00 6D 00 61 00 67 00  w.a.v.|.i.m.a.g.
10021800: 65 00 2F 00 74 00 69 00 66 00 66 00 7C 00 69 00  e./.t.i.f.f.|.i.
10021810: 6D 00 61 00 67 00 65 00 2F 00 78 00 2D 00 74 00  m.a.g.e./.x.-.t.
10021820: 69 00 66 00 66 00 7C 00 69 00 6D 00 61 00 67 00  i.f.f.|.i.m.a.g.
10021830: 65 00 2F 00 70 00 6E 00 67 00 7C 00 69 00 6D 00  e./.p.n.g.|.i.m.
10021840: 61 00 67 00 65 00 2F 00 78 00 2D 00 70 00 6E 00  a.g.e./.x.-.p.n.
10021850: 67 00 7C 00 69 00 6D 00 61 00 67 00 65 00 2F 00  g.|.i.m.a.g.e./.
10021860: 78 00 2D 00 74 00 61 00 72 00 67 00 61 00 7C 00  x.-.t.a.r.g.a.|.
10021870: 69 00 6D 00 61 00 67 00 65 00 2F 00 78 00 2D 00  i.m.a.g.e./.x.-.
10021880: 6D 00 61 00 63 00 70 00 61 00 69 00 6E 00 74 00  m.a.c.p.a.i.n.t.
10021890: 7C 00 69 00 6D 00 61 00 67 00 65 00 2F 00 78 00  |.i.m.a.g.e./.x.
100218A0: 2D 00 62 00 6D 00 70 00 7C 00 69 00 6D 00 61 00  -.b.m.p.|.i.m.a.
100218B0: 67 00 65 00 2F 00 78 00 2D 00 70 00 68 00 6F 00  g.e./.x.-.p.h.o.
100218C0: 74 00 6F 00 73 00 68 00 6F 00 70 00 7C 00 69 00  t.o.s.h.o.p.|.i.
100218D0: 6D 00 61 00 67 00 65 00 2F 00 78 00 2D 00 73 00  m.a.g.e./.x.-.s.
100218E0: 67 00 69 00 00 00 00 00 46 00 0F 00 01 00 4F 00  g.i.....F.....O.
100218F0: 72 00 69 00 67 00 69 00 6E 00 61 00 6C 00 46 00  r.i.g.i.n.a.l.F.
10021900: 69 00 6C 00 65 00 6E 00 61 00 6D 00 65 00 00 00  i.l.e.n.a.m.e...
10021910: 4E 00 50 00 51 00 54 00 50 00 4C 00 55 00 47 00  N.P.Q.T.P.L.U.G.
10021920: 49 00 4E 00 2E 00 44 00 4C 00 4C 00 00 00 00 00  I.N...D.L.L.....
```

Appendix
2

Appendix 2: Netscape Plug-In Information



Mime Type	Description	Suffixes	Enabled
image/x-sgi	SGI	sgi	Yes
image/x-photoshop	Photoshop	psd	Yes
image/x-bmp	BMP	bmp	Yes
image/x-macpaint	MacPaint	pnt, pntg	Yes
image/x-targa	Targa	tga, targa	Yes
image/x-png	PNG	png	Yes
image/png	PNG	png	Yes
image/x-tiff	TIFF	tif, tiff	Yes
image/tiff	TIFF	tif, tiff	Yes
audio/x-wav	WAV	wav	Yes
audio/wav	WAV	wav	Yes
audio/x-midi	MIDI	midi, mid	Yes
audio/midi	MIDI	midi, mid	Yes
video/flc	FLC	flc	Yes
video/avi	AVI	avi	Yes
video/x-msvideo	AVI	avi	Yes
audio/basic	AU	au, snd	Yes
audio/x-aiff	AIFF	aif	Yes
audio/aiff	AIFF	aiff	Yes
video/quicktime	QuickTime Movie	mov	Yes

Appendix

3

Appendix 3: Mindcraft's Simple Plug-In

The following are the changes (diffs) needed to turn Netscape's "WinTemplate" sample provided with the Netscape Plug-in Software Development Kit (available from ftp://ftp.netscape.com/pub/sdk/plug-in/windows/oct_21_97/winNPSDK.zip) into our test plug-in (we could not provide the integrated source code here because "WinTemplate" is copyrighted by Netscape):

```
27,29d26
<
<   NPMIMEType                  myMIMEType;  /*MC*/
<   char                        *src; /*MC*/
126,138d122
<   { /* BEGIN MC ADDED CODE */
<           int i;
<           This->myMIMEType = strdup(pluginType);
<           This->src = NULL;
<           for (i = 0; i < argc; i++)
<           {
<                   if (_stricmp(argn[i], "src") == 0)
<                   {
<                           This->src = strdup(argv[i]);
<                           break;
<                   }
<           }
<   } /* END MC ADDED CODE */
186,191d169
<   if (This != NULL)
<   {
<           if (This->src != NULL)
<                   free(This->src);
<   }
<
299,301d276
<   if (This->fMode == NP_FULL)
<           This->src = strdup(stream->url);
<
525,541c500
<                   /* MC: TextOut(hdc, 0, 0, '', Hello, World!'', 13 ); */
<
<
<                   { /* BEGIN MC ADDED CODE */
<                           RECT r;
<                           char outbuf[256];
<
<                           GetClientRect(hWnd, &r);
<
<                           sprintf(outbuf, "%s\n%s tag used\n%s",
<                                   This->myMIMEType,
```

```
<                                               (This->fMode == NP_EMBED) ? "EMBED"
: "HREF",
<                                               (This->src != NULL) ? This->src : "no SRC
attribute");
<
<                               DrawText(hdc, outbuf, strlen(outbuf), &r, DT_LEFT);
<                       } /* END MC ADDED CODE */
<
---
>                       TextOut(hdc, 0, 0, "Hello, World!", 13 );
```

Appendix

4

Appendix 4: Mindcraft's InstallShield QuickTime Plug-In Fix

```
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
// By using this sofware or source code you agree to the following License terms:   //
//                                                                            //
// Grant of License.                                                          //
//                                                                            //
// Mindcraft, Inc., the author of this software, grants you a personal,       //
// nontransferrable, nonexclusive, license to use this software on any computer     //
// system to which you personally have access.                                //
//                                                                            //
// Copyright.                                                                 //
//                                                                            //
// This software is copyright (C)1998 Mindcraft, Inc. All rights are reserved.      //
//                                                                            //
//                                                                            //
// LIMITATION OF LIABILITY.                                                   //
//                                                                            //
// UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING        //
// NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL MINDCRAFT BE LIABLE TO YOU OR ANY     //
// OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF  //
// ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK  //
// STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL       //
// DAMAGES OR LOSSES, EVEN IF MINDCRAFT SHALL HAVE BEEN INFORMED OF THE POSSIBILITY //
// OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR   //
// DEATH OR PERSONAL INJURY RESULTING FROM MINDCRAFT'S NEGLIGENCE TO THE EXTENT     //
// APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE    //
// EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT          //
// EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.                             //
//                                                                            //
// DISCLAIMER OF WARRANTY.                                                    //
//                                                                            //
// THIS CODE IS PROVIDED UNDER LICENSE ON AN ``AS IS'' BASIS, WITHOUT WARRANTY OF   //
// ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES //
// THAT THE CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR  //
// NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE  OF THE CODE IS //
// WITH YOU. SHOULD THE CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT MINDCRAFT)    //
// ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS           //
// DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF  //
// ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.           //
//                                                                            //
// U.S. GOVERNMENT END USERS.                                                 //
//                                                                            //
// The Covered Code is a ``commercial item,'' as that term is defined in 48 C.F.R.  //
// 2.101 (Oct. 1995), consisting of ``commercial computer software'' and            //
// ``commercial computer software documentation,'' as such terms are used in 48     //
// C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R.       //
// 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire //
// Covered Code with only those rights set forth herein.                      //
//                                                                            //
// MISCELLANEOUS.                                                             //
//                                                                            //
// This License represents the complete agreement concerning subject matter hereof. //
// If any provision of this License is held to be unenforceable, such provision     //
// shall be reformed only to the extent necessary to make it enforceable. This      //
// License shall be governed by California law provisions (except to the extent     //
// applicable law, if any, provides otherwise), excluding its conflict-of-law       //
// provisions. With respect to disputes in which at least one party is a citizen    //
```

```
// of, or an entity chartered or registered to do business in, the United States of //
// America: (a) unless otherwise agreed in writing, all disputes relating to this   //
// License (excepting any dispute relating to intellectual property rights) shall   //
// be subject to final and binding arbitration, with the losing party paying all    //
// costs of arbitration; (b) any arbitration relating to this Agreement shall be    //
// held in Santa Clara County, California, under the auspices of JAMS/ EndDispute;   //
// and (c) any litigation relating to this Agreement shall be subject to the        //
// jurisdiction of the Federal Courts of the Northern District of California, with  //
// venue lying in Santa Clara County, California, with the losing party responsible //
// for costs, including without limitation, court costs and reasonable attorneys    //
// fees and expenses. The application of the United Nations Convention on Contracts //
// for the International Sale of Goods is expressly excluded. Any law or regulation //
// which provides that the language of a contract shall be construed against the    //
// drafter shall not apply to this License.                                         //
//                                                                                  //
//////////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////////////////
//                                                                                  //
// InstallShield program to fix QuickTime plugin installation                        //
// Written by: Greg Burrell                                                         //
// Date: 10/30/98                                                                   //
//                                                                                  //
//////////////////////////////////////////////////////////////////////////////////////

// NOTE: If you are using InstallShield to build the demonstration then you
//       need the files setup.rul, README.txt, and license.txt which are
//       available at the Mindcraft web site (http://www.mindcraft.com/qtfix)

#include "sdlang.h"
#include "sddialog.h"

#define UNINST_LOGFILE_NAME      "Uninst.isu"
#define MIME_DB_CONTENTTYPE_REGKEY "MIME\\Database\\Content Type\\"
#define IE_REGKEY              "SOFTWARE\\Microsoft\\Internet Explorer\\"
#define IE_EXE_REGKEY          "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\App
Paths\\IEXPLORE.EXE"
#define IE_EXE_PROGNAME        "iexplore.exe"
#define IE_PLUGIN_FOLDER_NAME  "Plugins"

#define PLUGIN_DLL             "npqtplugin.dll"
#define PLUGIN_VERSION         "2.0.1"
#define PLUGIN_NAME            "QuickTime Plug-In"

#define TEST_PAGE              "http://www.apple.com/quicktime/samples/showcase/index.html"
#define LICENSE_FILENAME        "license.txt"

     // generated
     prototype MoveFileData();
     prototype HandleMoveDataError( NUMBER );
     prototype ProcessBeforeDataMove();
     prototype ProcessAfterDataMove();
     prototype SetupRegistry();
     prototype CleanUpInstall();
     prototype SetupInstall();
     prototype SetupScreen();
     prototype CheckRequirements();
     prototype DialogShowSdWelcome();
     prototype DialogShowSdLicense();
     prototype DialogShowSdFinishReboot();

     prototype MC_VerifyIEAndQTPlugin();
     prototype MC_GetIERegdata();
     prototype MC_ShowLicense();
     prototype MC_Tell_User_About_QT_Control_Panel();

     prototype MC_Disable_ActiveX_Precedence(STRING, STRING);
     prototype MC_Tell_IE_Plugin_Supports_Extension(STRING, STRING);
     prototype MC_Add_RegKey_If_Absent(STRING);
     prototype MC_Set_RegValue_If_Different(STRING, STRING, STRING);
```

```
  // ----- global variables ------

     BOOL        bWinNT, bIsShellExplorer, bInstallAborted, bIs32BitSetup;
     STRING      svDir;
     STRING      svName, svCompany, svSerial;
     STRING      svDefGroup;
     STRING      szAppPath;
     STRING      svSetupType;

     BOOL        bHasIE;
     STRING      svIEPluginExePath;
     STRING      svIEExePath;


////////////////////////////////////////////////////////////////////////////
//
//   MAIN PROGRAM
//
//      The setup begins here by hiding the visible setup
//      window.  This is done to allow all the titles, images, etc. to
//      be established before showing the main window.  The following
//      logic then performs the setup in a series of steps.
//
////////////////////////////////////////////////////////////////////////////
program
    Disable( BACKGROUND );

    CheckRequirements();

    SetupInstall();

    SetupScreen();

    if (DialogShowSdWelcome() != NEXT) goto end_install;

    MC_VerifyIEAndQTPlugin();

    if (DialogShowSdLicense() != NEXT) goto end_install;

    if (ProcessBeforeDataMove()<0) goto end_install;

    if (MoveFileData()<0) goto end_install;

    if (ProcessAfterDataMove()<0) goto end_install;

    if (SetupRegistry()<0) goto end_install;

  end_install:

    CleanUpInstall();

     // If an unrecoverable error occurred, clean up the partial installation.
     // Otherwise, exit normally.

    if (bInstallAborted) then
        abort;
    endif;

endprogram


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: DialogShowSdWelcome                                          //
//                                                                        //
//  Purpose: This function handles the standard welcome dialog.           //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function DialogShowSdWelcome()
    NUMBER nResult;
```

```
    STRING szTitle, szMsg;
 begin

  szTitle = "Welcome To the QuickTime for Internet Explorer Fix";
  szMsg   = "";
  nResult = SdWelcome(szTitle, szMsg);

  return nResult;
 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: MC_VerifyIEAndQTPlugin                                      //
//                                                                        //
//  Purpose: Explain to the user what this fix does and give the user a   //
//           chance to bail out.                                          //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function MC_VerifyIEAndQTPlugin()
  BOOL bvOpt1;
  BOOL bvOpt2;
begin
  // Fine IE and QT plug-in data in registry.
  MC_GetIERegdata();

  // Quit if couldn't find IE
  if (bHasIE != TRUE) then
      SdFinish("Unable to Find Internet Explorer",
               "Setup was unable to find Internet Explorer.",
               "No changes were made to this system.\n" +
                   "Click Finish to exit setup.",
               "",
               "",
               bvOpt1,
               bvOpt2);
      exit;
  endif;

  // Quit if couldn't find IE's Plugins folder.
  if (svIEPluginExePath = "") then
      SdFinish("Unable to Find Internet Explorer's Plugins Folder",
               "Setup was unable to find Internet Explorer's Plugins folder." +
                   "Please make sure that Internet Explorer was installed properly. ",
               "No changes were made to this system.\n" +
                   "Click Finish to exit setup.",
               "",
               "",
               bvOpt1,
               bvOpt2);
      exit;
  endif;

  // Quit if can't find QT plug-in in IE's Plugins folder
  if (Is(FILE_EXISTS, svIEPluginExePath) != TRUE) then
      SdFinish("QuickTime Plug-in Not Installed in Internet Explorer",
               "The QuickTime plug-in has not yet been installed in Internet Explorer." +
                   "Please install QuickTime and then run this fix again. ",
               "No changes were made to this system.\n" +
                   "Click Finish to exit setup.",
               "",
               "",
               bvOpt1,
               bvOpt2);
      exit;
  endif;
end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
```

```
// Function: MC_GetIERegdata                                                //
//                                                                          //
//  Purpose: Search the registry and see if Internet Explorer is installed  //
//           on the system. If so then find the path to its QT plug-in.     //
//           Finally, make sure the QT plug-in exists in IE's Plugins       //
//           folder.                                                        //
//                                                                          //
//////////////////////////////////////////////////////////////////////////////
function MC_GetIERegdata()
  NUMBER nRet;
  NUMBER nType;
  NUMBER nvSize;
  NUMBER nStrPos;
  STRING svAppMainPath;
begin
  if (RegDBSetDefaultRoot(HKEY_LOCAL_MACHINE) < 0) then
      MessageBox("Unable to access HKEY_LOCAL_MACHINE in registry", SEVERE);
      abort;
  endif;

  // To get IE Plugin directory:
  //    1. get exe path from registry.
  //    2. in exe path: find name of Internet Explorer executable.
  //    3. extract string up to "iexplore.exe"
  //    4. append "Plugins" since Plugins directory must
  //       be a subdirectory of the main app directory.
  bHasIE = FALSE;
  svIEPluginExePath = "";
  nRet = RegDBGetKeyValueEx(IE_EXE_REGKEY,
                            "",
                            nType,
                            svIEExePath,
                            nvSize);
  if ( (nRet = 0) && (nType = REGDB_STRING) && (svIEExePath != "") ) then
      nStrPos = StrFind(svIEExePath, IE_EXE_PROGNAME);
      if (nStrPos > 0) then
          StrSub(svAppMainPath, svIEExePath, 0, nStrPos);
          bHasIE = TRUE;
          svIEPluginExePath = svAppMainPath ^ IE_PLUGIN_FOLDER_NAME ^ PLUGIN_DLL;
      endif;
  endif;
end;


//////////////////////////////////////////////////////////////////////////////
//                                                                          //
// Function: DialogShowSdLicense                                            //
//                                                                          //
//  Purpose: This function displays the license agreement dialog.           //
//                                                                          //
//                                                                          //
//////////////////////////////////////////////////////////////////////////////
function DialogShowSdLicense()
    NUMBER nResult;
    STRING szTitle, szMsg, szQuestion, szLicenseFile;
 begin
  Disable(BACKBUTTON);
  szLicenseFile = SUPPORTDIR ^ LICENSE_FILENAME;
  szTitle    = "";
  szMsg      = "";
  szQuestion = "";
  nResult    = SdLicense( szTitle, szMsg, szQuestion, szLicenseFile );

  return nResult;
 end;


//////////////////////////////////////////////////////////////////////////////
//                                                                          //
// Function: ProcessBeforeDataMove                                          //
//                                                                          //
```

```
//  Purpose: This function performs any necessary operations prior to the    //
//           actual data move operation.                                     //
//                                                                           //
///////////////////////////////////////////////////////////////////////////////
function ProcessBeforeDataMove()
    STRING svLogFile;
    NUMBER nResult;
 begin

  InstallationInfo( @COMPANY_NAME, @PRODUCT_NAME, @PRODUCT_VERSION, @PRODUCT_KEY );

  svLogFile = UNINST_LOGFILE_NAME;

  nResult = DeinstallStart( svDir, svLogFile, @UNINST_KEY, 0 );
  if (nResult < 0) then
      MessageBox( @ERROR_UNINSTSETUP, WARNING );
  endif;

  szAppPath = TARGETDIR;

  if ((bIs32BitSetup) && (bIsShellExplorer)) then
      RegDBSetItem( REGDB_APPPATH, szAppPath );
      RegDBSetItem( REGDB_APPPATH_DEFAULT, szAppPath ^ @PRODUCT_KEY );
      RegDBSetItem( REGDB_UNINSTALL_NAME, @UNINST_DISPLAY_NAME );
  endif;

  return 0;
 end;

///////////////////////////////////////////////////////////////////////////////
//                                                                           //
// Function:  MoveFileData                                                   //
//                                                                           //
//  Purpose:  This function handles the data movement for                    //
//            the setup.                                                     //
//                                                                           //
///////////////////////////////////////////////////////////////////////////////
function MoveFileData()
    NUMBER nResult, nDisk;
 begin

  nDisk = 1;
  SetStatusWindow( 0, "" );
  Disable( DIALOGCACHE );
  Enable( STATUS );
  StatusUpdate( ON, 100 );
  nResult = ComponentMoveData( MEDIA, nDisk, 0 );

  HandleMoveDataError( nResult );

  Disable( STATUS );

  return nResult;

 end;


///////////////////////////////////////////////////////////////////////////////
//                                                                           //
// Function: HandleMoveDataError                                             //
//                                                                           //
//  Purpose: This function handles the error (if any) during the move data   //
//           operation.                                                      //
//                                                                           //
///////////////////////////////////////////////////////////////////////////////
function HandleMoveDataError( nResult )
 begin

  switch (nResult)
  case 0:
       return 0;
```

```
  default:
       SprintfBox( SEVERE, @TITLE_CAPTIONBAR, @ERROR_MOVEDATA,  nResult );
       bInstallAborted = TRUE;
       return nResult;
  endswitch;

 end;

////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: ProcessAfterDataMove                                         //
//                                                                        //
//  Purpose: This function performs any necessary operations needed after //
//           all data has been moved.                                     //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function ProcessAfterDataMove()
    STRING szReferenceFile;
 begin

  // DeinstallSetReference specifies a file to be checked before
  // uninstallation. If the file is in use, uninstallation will not proceed.

  szReferenceFile = svDir ^ @PRODUCT_KEY;
  DeinstallSetReference( szReferenceFile );


  return 0;
 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: CleanUpInstall                                               //
//                                                                        //
//  Purpose: This cleans up the setup.  Anything that should              //
//           be released or deleted at the end of the setup should        //
//           be done here.                                                //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function CleanUpInstall()
 begin


  if (bInstallAborted) then
      return 0;
  endif;

  DialogShowSdFinishReboot();

  if (BATCH_INSTALL) then // ensure locked files are properly written
      CommitSharedFiles(0);
  endif;

  return 0;
 end;

////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: SetupInstall                                                 //
//                                                                        //
//  Purpose: This will setup the installation.  Any general initialization //
//           needed for the installation should be performed here.        //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function SetupInstall()
 begin

  Enable( CORECOMPONENTHANDLING );

  bInstallAborted = FALSE;
```

```
  if (bIs32BitSetup) then
      svDir = PROGRAMFILES ^ @COMPANY_NAME ^ @PRODUCT_NAME;
  else
      svDir = PROGRAMFILES ^ @COMPANY_NAME16 ^ @PRODUCT_NAME16;
  endif;

  TARGETDIR  = svDir;

  SdProductName( @PRODUCT_NAME );

  Enable( DIALOGCACHE );

  return 0;
 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function:  SetupScreen                                                 //
//                                                                        //
//  Purpose:  This function establishes  the screen look.  This includes  //
//            colors, fonts, and text to be displayed.                    //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function SetupScreen()
 begin
   SetColor(BACKGROUND, BK_BLUE | BK_SMOOTH);
   Enable( FULLWINDOWMODE );
   SetTitle( @TITLE_MAIN, 24, WHITE );
   SetTitle( @TITLE_CAPTIONBAR, 0, BACKGROUNDCAPTION ); // Caption bar text.
   Enable( BACKGROUND );
   Delay( 1 );
 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function:  CheckRequirements                                           //
//                                                                        //
//  Purpose:  This function checks all minimum requirements for the       //
//            application being installed.  If any fail, then the user    //
//            is informed and the setup is terminated.                    //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function CheckRequirements()
    NUMBER  nvDx, nvDy, nvResult;
    STRING  svResult;

 begin

  bWinNT           = FALSE;
  bIsShellExplorer = FALSE;

  // Check screen resolution.
  GetExtents( nvDx, nvDy );

  if (nvDy < 480) then
      MessageBox( @ERROR_VGARESOLUTION, WARNING );
      abort;
  endif;

  // set 'setup' operation mode
  bIs32BitSetup = TRUE;
  GetSystemInfo( ISTYPE, nvResult, svResult );
  if (nvResult = 16) then
      bIs32BitSetup = FALSE; // running 16-bit setup
      return 0; // no additional information required
  endif;

  // --- 32-bit testing after this point ---
```

```
  // Determine the target system's operating system.
  GetSystemInfo( OS, nvResult, svResult );

  if (nvResult =  IS_WINDOWSNT) then
      // Running Windows NT.
      bWinNT = TRUE;

      // Check to see if the shell being used is EXPLORER shell.
      if (GetSystemInfo( OSMAJOR, nvResult, svResult ) = 0) then
          if (nvResult >= 4) then
              bIsShellExplorer = TRUE;
          endif;
      endif;

  elseif (nvResult = IS_WINDOWS95 ) then
      bIsShellExplorer = TRUE;

  endif;

 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: MC_Tell_User_About_QT_Control_Panel                          //
//                                                                        //
//  Purpose:  Tell the user to set all three "File Type Associations"     //
//            check boxes in the QuickTime control panel.                 //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function MC_Tell_User_About_QT_Control_Panel()
begin
  MessageBox("In order for this fix to have full effect you must\n" +
             "go to the File Type Associations setting in the QuickTime\n" +
             " control panel and set all three checkboxes.",
             INFORMATION);
end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: DialogShowSdFinishReboot                                     //
//                                                                        //
//  Purpose: This function will show the last dialog of the product.      //
//           It will allow the user to reboot and/or show some readme text.  //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function DialogShowSdFinishReboot()
    NUMBER nResult, nDefOptions;
    STRING szTitle, szMsg1, szMsg2, szOption1, szOption2;
    NUMBER bOpt1, bOpt2;
    STRING svCmdLine;
 begin

  MC_Tell_User_About_QT_Control_Panel();

  szTitle = "Fix Installation Complete";
  szMsg1 = "The QuickTime Plugin fix for Internet Explorer has been installed.";
  szMsg2 = "";

  if (!BATCH_INSTALL) then
      bOpt1 = FALSE;
      bOpt2 = FALSE;
      szOption1 = "Start Internet Explorer with QuickTime samples web page.";
      szOption2 = "";
      nResult = SdFinish( szTitle, szMsg1, szMsg2, szOption1, szOption2, bOpt1, bOpt2 );

      // If the user chose then launch Internet Explorer with our test page.
      if (bOpt1 = TRUE) then
          svCmdLine = "-nohome " + TEST_PAGE;
```

```
          LaunchApp(svIEExePath, svCmdLine);
      endif;

      return 0;
  endif;

  nDefOptions = SYS_BOOTMACHINE;
  nResult     = SdFinishReboot( szTitle, szMsg1, nDefOptions, szMsg2, 0 );

  return nResult;
 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: SetupRegistry                                                //
//                                                                        //
//  Purpose: This function makes the registry entries for this fix.       //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function SetupRegistry()
  STRING szIEDisableRegistryInitRegkey;
begin
  MC_Tell_IE_Plugin_Supports_Extension("video/quicktime",   ".qt");
  MC_Tell_IE_Plugin_Supports_Extension("audio/aiff",        ".aifc");
  MC_Tell_IE_Plugin_Supports_Extension("video/x-msvideo",   ".vfw");

  // Now set the Plugins\DisableRegistryInitOnStartup registry key to true.
  // Otherwise IE will remove the keys we added to Plugins\Extension.
  szIEDisableRegistryInitRegkey = IE_REGKEY + "Plugins\\DisableRegistryInitOnStartup";
  MC_Add_RegKey_If_Absent(szIEDisableRegistryInitRegkey);
  MC_Set_RegValue_If_Different(szIEDisableRegistryInitRegkey, "", "true");

  MC_Disable_ActiveX_Precedence("audio/aiff",        ".aifc");
  MC_Disable_ActiveX_Precedence("video/x-msvideo",   ".vfw");

  return 0;
 end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: MC_Tell_IE_Plugin_Supports_Extension                         //
//                                                                        //
//           Add the extension to IE's Plugins registry key.              //
//                                                                        //
//           A plugin DLL must contain a FileExtents line in its version  //
//           information.  FileExtents lists all the file extensions that //
//           the plugin supports.  For each of these filename extensions  //
//           the Internet Explorer web broswer creates a key under the    //
//           HKLM\SOFTWARE\Microsoft\Internet Explorer\Plugins\Extension  //
//           key.  The values it contains are "default", "Content Type",  //
//           "Version", and "Location".  This is how the IE web browser   //
//           knows what plugin to use for what filename extensions.  If a //
//           plugin DLL omits a filename extension then we can still add  //
//           the registry key so that the IE browser knows it should use  //
//           the plugin for files with that filename extension.           //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function MC_Tell_IE_Plugin_Supports_Extension(svMimetype, svExtension)
  STRING svIEPluginExtensionRegkey;
begin
  if (RegDBSetDefaultRoot(HKEY_LOCAL_MACHINE) < 0) then
      MessageBox("Unable to access HKEY_LOCAL_MACHINE in registry", SEVERE);
      abort;
  endif;

  svIEPluginExtensionRegkey = IE_REGKEY + "Plugins\\Extension\\" + svExtension;

  MC_Add_RegKey_If_Absent(svIEPluginExtensionRegkey);
```

```
  MC_Set_RegValue_If_Different(svIEPluginExtensionRegkey, ""            , PLUGIN_NAME);
  MC_Set_RegValue_If_Different(svIEPluginExtensionRegkey, "Content Type",  svMimetype);
  MC_Set_RegValue_If_Different(svIEPluginExtensionRegkey, "Version"     , PLUGIN_VERSION);
  MC_Set_RegValue_If_Different(svIEPluginExtensionRegkey, "Location"    ,  svIEPluginExePath);
end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: MC_Disable_ActiveX_Precedence                                //
//                                                                        //
//           If the mimetype already exists in the registry and has a     //
//           CLSID, then add the extension the EnablePlugin subsection    //
//           of that CLSID.                                               //
//                                                                        //
//           Each ActiveX control is identified in the registry by a CLSID //
//           (HKCR\CLSID\<CLSID>).  A CLSID may have a sub-key named      //
//           "EnablePlugin".  Each key under EnablePlugin is named for a  //
//           filename extension (for example, "EnablePlugin\.qt").  For   //
//           those filename extensions, that particular ActiveX control   //
//           will not have precedence over plugins installed in the       //
//           Internet Explorer web browser.                               //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function MC_Disable_ActiveX_Precedence(svMimetype, svExtension)
  STRING svMimeRegkey;
  STRING svCLSIDValue;
  NUMBER nRet;
  NUMBER nType;
  NUMBER nvSize;
begin
  if (RegDBSetDefaultRoot(HKEY_CLASSES_ROOT) < 0) then
      MessageBox("Unable to access HKEY_CLASSES_ROOT in registry", SEVERE);
      abort;
  endif;

  svMimeRegkey = MIME_DB_CONTENTTYPE_REGKEY +  svMimetype;
  if (RegDBKeyExist(svMimeRegkey) = 1) then
      nRet = RegDBGetKeyValueEx(svMimeRegkey,
                                "CLSID",
                                nType,
                                svCLSIDValue,
                                nvSize);
      if ( (nRet = 0) && (nType = REGDB_STRING) && (svCLSIDValue != "") ) then
          MC_Add_RegKey_If_Absent("CLSID\\" + svCLSIDValue + "\\EnablePlugin\\" +
svExtension);
      endif;
  endif;
end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
// Function: MC_Add_RegKey_If_Absent                                      //
//                                                                        //
//  Purpose: If the given key isn't in the registry then add it.          //
//                                                                        //
////////////////////////////////////////////////////////////////////////////
function MC_Add_RegKey_If_Absent(szRegKey)
begin
  if (RegDBKeyExist(szRegKey) != 1) then
      if (RegDBCreateKeyEx(szRegKey, "") < 0) then
          MessageBox("Unable to create registry key: " + szRegKey, SEVERE);
          abort;
      endif;
  endif;
end;


////////////////////////////////////////////////////////////////////////////
//                                                                        //
```

```
// Function: MC_Set_RegValue_If_Different                                    //
//                                                                           //
//  Purpose: Set a value for a given reg key.  Only set it if this new       //
//           value differs from the current value in the registry.           //
//                                                                           //
///////////////////////////////////////////////////////////////////////////////
function MC_Set_RegValue_If_Different(szRegKey, szName, szValue)
  NUMBER nvType;
  STRING svOldValue;
  NUMBER nvSize;
  NUMBER nRet;
begin
  nRet = RegDBGetKeyValueEx(szRegKey,
                            szName,
                            nvType,
                            svOldValue,
                            nvSize);
  if ( (nRet != 0) || (svOldValue != szValue) ) then
      // either the name wasn't present in the regkey or its old value
      // differs from this new value.  So go ahead and set the new value.
      if (RegDBSetKeyValueEx(szRegKey,
                             szName,
                             REGDB_STRING,
                             szValue,
                             -1) < 0) then
          MessageBox("Unable to create registry value: " + szRegKey, SEVERE);
          abort;
      endif;
  endif;
end;


#include "sddialog.rul"
```

Appendix
5

Appendix 5: Apple's Use of the EnablePlugin Registry Key

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.wav

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.midi

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.mid

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.flc

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.avi

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.au

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.snd

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.aif

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.aiff

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.mov

HKCR\CLSID\{05589FA1-C356-11CE-BF01-00AA0055595A}\EnablePlugin\.qt

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.aiff

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.au

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.wav

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.aiff

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.wav

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.avi

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.mov

HKCR\CLSID\{05589fa1-c356-11ce-bf01-00aa0055595a}\EnablePlugin\.avi