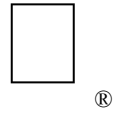


New Technical Notes

Macintosh



Developer Support

Apple's Multidisk Installer

Platforms & Tools

Revised by: Rich Kubota
Written by: scott douglass

January 1992
March 1986

This Technical Note documents Apple's Multidisk Installer, and it is in addition to separate Installer documentation which provides the details of writing scripts.

Changes since September 1991: Revised information on the use of Installer version 3.1 to version 3.2. Revised information on the use of ScriptCheck version 3.2.1 with Installer version 3.2. Added Common Questions and Answers relating to the use of the Installer.

Apple's Multidisk Installer is intended to make it easy for Macintosh users to add or update software. It is a very useful tool for adding third-party software, and Apple recommends that you use the Installer unless your software installation is simple. Apple also recommends that you use version 3.2 of the Installer.

The Multidisk Installer has the following features, as of version 3.2:

- "Easy Install" mode where the Installer script writer can determine the appropriate installation based upon examination of the target environment and provide the user with "One-Button Installation"
- An optional "Custom Install" mode where power users can customize their installation
- "Live" installation to the currently booted and active system; thus it is no longer necessary to ship the Installer on a bootable disk with a System Folder
- Ability to install from an AppleShare server ("Network Install")
- Ability to install from multiple source disks
- Installation of software to folders other than the System Folder as well as creation of new folders as necessary
- Runs under System 4.2 and later versions
- "User Function" support; this feature provides linkage to developer-defined code segments during Easy Install, so script writers can customize the process of determining what software to install and how to install it
- "Action Atom" support; this feature provides linkage to developer-defined code segments that are called before or after the installation takes place; script writers can use this feature to extend the capabilities of the Installer
- Audit Records; this feature provides the script writer with the ability to record

details about an installation so that future installations can be more intelligent

The `'indm'` (default map) resource of Installer 3.0.1 is no longer supported in Installer 3.1 and later versions. This was used by script writers to implement Easy Install. It is replaced by `'infr'` (framework) and `'inrl'` (rule) resources.

Note: If the user opens the Installer document rather than the Installer, the wrong Installer may be launched (depending upon the contents of their mounted volumes). This is only a problem between versions 3.1 and 3.0.x. If you are developing a 3.1 script, you may want to add an 'indm' resource that puts up a warning dialog box. If you are developing a 3.0.x script, you may want to add an 'infr' and 'inrl' resource that puts up a `reportSysError` dialog box. This problem is resolved in Installer 3.2. With version 3.2, the file type and creator are both 'bjbc' as opposed to the use of 'cfbj' with versions 3.0.1 and 3.1.

Installer version 3.2 is available as a complete reference suite which includes the following:

- *Installer 3.2 Scripting Guide* (dated December 1, 1991, on the cover)
- *Installer ScriptCheck 3.2b7 User's Manual*
- Installer 3.2 application
- ScriptCheck 3.2.1 (MPW Tool)
- InstallerTypes.r (MPW Rez interface file)
- ActionAtomIntf.a, .h, .p (Action Atom interface files for Assembler, C, and Pascal)

The reference suite for Multidisk Installer 3.2 is available on the latest Developer CD and on AppleLink in the Developer Services Bulletin Board. The Multidisk Installer was also provided on the System 7 Golden Master CD-ROM; however, that package included the b7 release of the MPW ScriptCheck tool.

Multidisk Installer version 3.2 contains a few minor improvements that will make it easier to write scripts that work on both System 6.0.x and 7.0. Installer 3.1 had minimal testing with System 7.0. If you are expecting to install software onto machines running System 7.0, you should consider upgrading. Script changes should be minimal.

Common Question and Answers

Q How can I check for a minimum system version?

A Use the `checkFileVersion` clause as part of the 'inrl' Rules Framework resource. The format of the minimal-version parameter is shown in the `InstallerTypes.r` file as `#define Version'`. The most common difficulties are in remembering that BCD values are required and in dealing with two-digit version numbers. Some samples follow.

Assuming that the target-filespec resource, 'infs', for the System file is 1000, use the following clause to check for System version 6.0.5:

```
checkFileVersion{1000, 6, 5, release, 0};
```

Assuming that the target-filespec resource, 'infs', for the Finder file is 1001, use the following clause to check for Finder version 6.1.5:

```
checkFileVersion{1001, 6, 0x15, release, 0};
```

Assuming that the target-filespec resource, 'infs', for the AppleTalk resource file is 1002, use the following clause to check for AppleTalk version 53:

```
checkFileVersion{1002, 0x53, 0, release, 0};
```

Q My Installer script installs a desk accessory. Under System 6, each time I run the script, a new copy of the DA appears as a DRVR resource in the System file. Why?

A Unfortunately, this is a symptom when the 'deleteWhenInstalling' flag is used in conjunction with the 'updateExisting' flag. The *Installer 3.1 & 3.2 Scripting Guide* indicates that resources marked with the 'dontDeleteWhenInstalling' flag can be replaced with a new resource. The guide also indicates that the Installer will overwrite a preexisting resource in the target file if the 'updateExisting' flag is set. Given these two flag settings, and the use of the replace 'byName' (noByID) flag, the Installer does not delete the DA. Instead a new DRVR resource is created with the same name but a new resource ID.

The correct Installer action is accomplished by setting the 'deleteWhenInstalling' flag in conjunction with the 'updateExisting' flag. Alternatively, use the 'dontDeleteWhenInstalling' flag with the 'keepExisting' flag.

Q How can I include the current volume name in a reportVolError alert as many of the installation scripts from Apple do?

A The volume name can be included by inserting “^0” in the desired location of the Pascal string passed to the reportVolError error reporting clause.

Q. I set the searchForFile flag in my 'infs' resource, however, the Installer acts as if it's unable to find the file. Why?

A. The likely reason for this problem is that the desired file is within a folder by the same name. When the searchForFile flag is set, the Installer will also find a match on a folder. The Installer will not replace a folder with a file, nor will it add a resource to a folder. The Installer continues as if the search failed.

Q What is the 'incd' resource about?

A When the MPW ScriptCheck tool is used, it reads the script's file creation date/time stamp and converts it into a long word with the Date2Secs procedure. ScriptCheck stores this long word in the 'incd' resource for use with verifying files when a network installation is performed. See the following questions for a discussion of this resource.

Q What checks are made by the Installer when preflighting an installation? Occasionally the alert "Could not find a required file . . ." occurs and the installation is aborted.

A The Installer compiles a list of the source file specifications from each of the resource 'inra' and file 'infa' atoms specified among the package 'inpk' atoms included for installation. Each source file specification includes a complete path name. As each source file is accessed, a check is made of the file's creation date/time stamp with the date/time stamp recorded in the corresponding 'infs' resource. If the date/time stamps

do not match, the alert results and the installation is aborted. The creation date/time stamp in the 'infs' resource can be

- entered manually into the script file so long as the value is not 1 or 0,
- filled in by ScriptCheck automatically, if a value of 1 is entered in the date field,
- forced to be updated, if the -d switch is used with ScriptCheck.

Q What are some of the considerations when configuring a network installation setup?

A Under Installer 3.1/3.2, network software installations are made possible by setting up an installation folder on the server volume. This folder will contain the Installer application, the Script file, and a folder(s) matching the names of the required disk(s). Within the disk folder(s) are the corresponding contents of the disk(s).

A problem can occur when a workstation is used to create the server installation folder and the system date and time differ significantly between the two systems. Under such condition, files copied from the workstation to the server may have their creation and modification date time/stamps altered. If a modification is made, the “delta” is constant for both the creation and modification date/time stamp and for all files copied at that time.

As indicated in the previous question, the Installer preflights a file by comparing its creation date/time stamp with the value stored in the corresponding 'infs' resource in the script file. To compensate for the fact that a server may alter a file's creation date/time stamp, the Installer implements the 'incd' resource. After the user selects the Install button, the Installer reads the 'incd' resource and compares it with the script file's creation date/time stamp. The difference is stored as the “delta.” On a normal disk installation, the “delta” is always zero. As the Installer finds each required source file, the file's creation date/time stamp is converted to a long word and adjusted by the “delta.” The modified date/time stamp is then compared with that stored in the script file. If the values match, the file is considered found and the installation proceeds. On network installations, the delta may be nonzero. If so, it indicates that the file's creation date/time stamps were modified when copied to the server. Thus the 'incd' resource gives the Installer a way to maintain file verification even though the date/time stamp may be altered.

A specific problem can occur when an installation is set up on some systems running older versions of Novell Server software. Under specific conditions, files copied to some Novell servers have their creation time stamp altered to 12:00 A.M. regardless of the original time stamp. This includes the creation time stamp of the script file. This condition wreaks havoc with the Installer's preflight mechanism. The “delta” determined between the 'incd' resource and the Script file's creation date/time stamp may not be consistent with the creation date/time stamp stored in the infs resource and the corresponding file's time stamp now at 12:00 A.M.

A work-around solution for this problem is to set the Creation time stamp for all files on the installation disk to 12:00 A.M. , BEFORE running the ScriptCheck tool. Use the MPW tool SetFile to perform this function. Here's a sample MPW script for performing this function:

```
SetFile    -d "1/1/92 12:00AM" `files -r -s -f ≈`
```

This script assumes that the current directory is set to the root of the Installation disk. For multiple disks, run this script on each disk. Use the '-f' switch with ScriptCheck to ensure that the date/time stamps are updated on scripts previously checked.

Installation of software is a nontrivial process. Apple recommends that you reserve time for development and testing to ensure that the installation process proceeds smoothly on all target machine configurations.

To ship the Installer with your product you must contact Apple's Software Licensing Department (AppleLink: SW.LICENSE) and license the Installer alone or with the system software package that includes the version of the Installer you intend to use. Software Licensing also supplies you with a copy of the Installer that you may ship.

Further Reference:

- Installer 3.2 Reference Suite