

New Technical Notes

Macintosh



®

Developer Support

Installer Q&As

Platform & Tools

M.PT.Installer.Q&As

Revised by: Developer Support Center

October 1992

Written by: Developer Support Center

October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you’ve sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don’t have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

|New Q&As and Q&As revised this month are marked with a bar in the side margin.

Macintosh Installer script references

Written: 4/11/91

Last reviewed: 6/14/91

I’m having problems creating an Installer script using the information in the Macintosh Technical Note "Apple's Multidisk Installer." Why can't I get Rez or SAREz to work with the sample script in that Note?

—

The problem may be that you’re using old Installer information. The installer and its interface files have been completely revised. If your version of this Technical Note doesn’t say “Revised October 1990,” you’re referencing old information.

The new information is available on AppleLink and on the current Developer CD Series disc and the System 7.0 CD (Development Tools:Installer folder). On the CDs you’ll find a complete toolkit for writing installer scripts, including:

- The ScriptWriter’s Guide

- Interface files
- Script Checker application
- Sample scripts

X-Ref:

Macintosh Technical Note “Apple’s Multidisk Installer”

System 7.0 InstallInit.r sample script Types.h correction

Written: 5/30/91

Last reviewed: 8/1/91

When I Rez the InstallInit.r sample script which comes with the Installer package on the Macintosh System 7.0 Golden Master CD, I encounter errors referencing the Types.h file. What’s going on?

—

This sample script was released without being edited to remove the reference to include the GestaltEqu.h file. In an early stage of System 7.0, the GestaltEqu.h consisted solely of #define statements which were useful in the CheckGestalt clause. With System 7.0b1, the GestaltEqu.h file was modified to include prototypes for the Gestalt, NewGestalt, and ReplaceGestalt functions. These prototypes required the inclusion of the Types.h file, including typedef statements that Rez does not understand. To correct this problem, simply delete the “include GestaltEqu.h” statement. Recent samples have been modified to correct this error.

Macintosh Installer 3.1 makes backup copies of active files

Written: 7/10/91

Last reviewed: 8/15/91

If I start up from an 800K floppy disk with only the Macintosh System and Finder files living on it (leaving some 350K of space free), and then try to use Installer 3.2 to move a 130K cdev and an 8K 'adbs' resource, I get a message that not enough disk space is available on the startup floppy disk. Installer says I need some 505K of space available, yet I am copying less than 150K. Why is this happening?

—

This results from the “live install” feature of the Installer. Because the floppy disk System file will be written to in this scenario, the Installer first attempts to make a backup copy of the system. This means that there needs to be additional disk space on the startup disk for a copy of all active files—in this case, the System file on the floppy drive. For the floppy there is insufficient room. If the system is started up from a different disk, the Installer installs the new resources into the inactive System file on a floppy disk without creating a backup.

How Macintosh Installer locates source files

Written: 8/28/91

Last reviewed: 10/1/91

I notice that if I change the name of the installation disk, the Installer is still able to find the source files. How does this work? I'm writing an action atom code resource and I want to duplicate the source file-finding scheme implemented by the Installer, as described on page 30 of the "Installer 3.1/3.2 Scripting Guide."

—

The Installer takes the source file spec and parses off the volume name leaving the filepath from the root of the current volume. It then searches for the filespec using this parsed file name from the root of the current volume. If the file is not found, then the search order continues with guidelines 1–4. In this manner, the Installer is able to install from disk copies of Installer disks when the diskette name has been changed.

Installer 3.2 and preserving a file's "locked" bit

Written: 8/28/91

Last reviewed: 9/24/91

When using the Apple Installer 3.2, source files which are locked become unlocked when the installer copies them to their destination. The installer preserves some of the other file attribute bits but not the locked bit. How can I preserve the locked bit? Will I have to write an action atom?

—

It appears that this is not a supported feature of the Installer. The problem here is that the locked bit is a Finder information setting. Were you to have your own Finder comments on a file source, you would find that these comments would no longer be part of the "installed" file.

In contrast, the Finder is designed to preserve this information during a copy. Apparently, the only method for preserving the locked bit is to do so through a post-installation action atom. You might use the target file 'infs' resource ID as the selector to the code resource, so that you can load the resource to determine the target file path.

Installer and finding a source file by type and creator

Written: 11/27/91

Last reviewed: 12/12/91

Can a Macintosh Installer script be written so a source file is found by its file and creator type rather than by name?

—

No. The Installer is not designed to find a source file by type and creator. However, we've passed your suggestion along to the the Installer engineering team.

Macintosh Installer and Apple event support

Written: 12/10/91

Last reviewed: 1/27/92

Does the Macintosh Installer support Apple events? We would like to launch Installer and

have it install automatically when we send it the appropriate Apple event.

—

The current Installer does not support Apple events; however, support for Apple events is being considered for a future version of the Installer.

Files for Installer post-installation action atom

Written: 12/10/91

Last reviewed: 1/27/92

When a post-installation action atom gets control, can we assume that all of the files the atom will work with are on the hard disk?

You may assume that all of the file and resource atoms included in the package resources have been fired and all files and resources designated for installation by these atoms have been copied to the target disk.

How Installer computes target volume storage size requirement

Written: 12/19/91

Last reviewed: 1/27/92

How does the Macintosh Installer compute the storage size requirement for a target volume?

The Installer creates a list of all files and resources which are to be installed. It then goes to each corresponding 'inra' and 'infa' resource, and sums up the size fields. If a live install is being performed, a check is also made to determine the size of active files, such as the System file, that need to be duplicated. The sizes of such files are added to the installation size requirement. For this reason, it's difficult to perform a live install onto a floppy startup disk, even if for a small resource. There is generally insufficient floppy disk space to make a copy of the active System file.

Macintosh resources: Dynamic installation and deletion

Written: 1/23/92

Last reviewed: 2/28/92

I've written an Installer script which has an 'infa' file atom to copy a new file, and then uses 'inra' resource atoms to delete resources from that file. When I finish the installation process and open the file with ResEdit, the file still contains the resources designated for deletion. Why?

When performing an installation, the Installer performs these general steps:

1. The Installer identifies all the resource and file atoms to be installed and determines whether there is sufficient target disk space. The Installer also takes into consideration that files designated for deletion will be copied into the temporary folder, (for live installations

only) so that the installation can be undone.

2. Pre-installation action atoms are run.

2. The Installer deletes all specified resources and files, (or moves them into the temporary folder for live installs), plus asks the user whether newer files will be replaced. After performing this function, the installer assumes all deletions are finished.

3. The Installer performs the installation of all designated files and resources.

The Installer handles the resource deletions first. Since the file doesn't exist or is already deleted, the deletion resource atoms are considered successful. The Installer assumes the installed files are already set as needed; that there's no need to delete resources from the file. As a result, resources, specified for deletion on a newly installed file, aren't deleted.

Bypassing Macintosh Installer's creation date check

Written: 1/23/92

Last reviewed: 2/28/92

How can I get the Macintosh Installer to bypass checking the creation date of source files when preflighting an installation?

—

Normally, the Installer checks that the source files exist by the name specified in the 'infs' resource and that the creation date/time stamp matches that specified in the resource as well. To bypass the second check, set the creation date field to zero. In addition, do **not** use the -d switch with ScriptCheck, which forces updating of the date field.

Macintosh Installer 3.2 & 3.3 live installation handling

Written: 3/2/92

Last reviewed: 4/7/92

I'm writing a pre-installation action atom to save some information from an existing Macintosh file which my script will later replace. When I do a live install, the file appears to have already been deleted. What's happening?

—

Installers 3.2 and 3.3 treat live installations differently from installations to non-boot volumes. To be able to restore the boot volume, should the installation be cancelled or be unsuccessful, the Installer moves all files designated for replacement, deletion, or modification into the Installer Temp folder in the System Folder. The pre-installation action atom can make use of two parameters passed in the parameter block to access files in the temporary folder. The "didLiveUpdate" flag indicates whether a live installation has been selected. The "installerTempDirID" is the DirID of the temporary folder. Your action atom code resource should check the "didLiveUpdate" flag. If true, then search for the desired file in the desired folder specified by "installerTempDirID".

Use pre-installation action atom to protect user prefs file

Written: 2/25/92

Last reviewed: 4/7/92

How can I keep the Macintosh Installer script from replacing a newer file? The various flag settings which I've tried always result in the target file being replaced by the source file.

—

The Installer uses the creation data/time stamp (as opposed to the modification date/time stamp) to determine whether the target file is newer than the source file. If the target file has the same or earlier date/time stamp as does the source file on the installation diskette, the file is replaced.

To control the installation of a file based on modification date requires an action atom. The function of the action atom code resource could range from halting the installation if the target file is greater than the source file, to performing the actual file copy. Unfortunately, there isn't a method for simply bypassing the installation of a single file or package. The Installer engineering group is looking into ways of handling this feature for a future release of the Installer.

Macintosh Installer preflight checks

Written: 9/30/91

Last reviewed: 10/1/91

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.

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.

Macintosh 'incd' resource

Written: 9/30/91

Last reviewed: 10/1/91

What is the 'incd' resource about?

When the MPW ScriptCheck tool is used, it reads the script's file creation date/time stamp converting 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.

Including current Mac volume name with reportVolError alert

Written: 9/30/91

Last reviewed: 10/1/91

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

—

The volume name can be included by inserting “^0” as part of the Pascal string passed to the reportVolError error reporting clause.

Why multiple 'DRVR' resources for DA installation

Written: 9/30/91

Last reviewed: 10/1/91

My Macintosh 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?

—

This happens when the `dontDeleteWhenInstalling` 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. On the other hand, use the `dontDeleteWhenInstalling` flag with the `keepExisting` flag.

checkFileVersion clause and 'inrl' resource

Written: 9/30/91

Last reviewed: 10/1/91

How can I check for the existence of a minimum Macintosh file version?

—

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 how to deal with two-digit version numbers. Some samples follow.

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

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

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

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

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

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

Network installation setup considerations

Written: 9/30/91

Last reviewed: 10/1/91

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

—

Under Installer 3.1 or 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).

One of the engineering problems which needed to be dealt with, can occur when a workstation is used to create the server installation folder. The problem occurs when the system date and time differ significantly between the workstation and the server. Under such a 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.

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 AM 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 files time stamp now at 12:00 AM.

A workaround solution for this problem is to set the Creation time stamp for all files on the installation disk to 12:00 AM, 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 "1/1/91 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.