Installer SDK
Overview

Document Version 1.2.3 - December 18, 1998

Table of Contents

Installer SDK

What's New?

As before, the goal of this SDK is to provide the latest and greatest Installer tools and helpful utilities, as well as to give you a look at where we're headed with the tools and the Installer itself. Look for the UPDATED! labels to help you discover the changes from the last SDK.

Scripting Support and Bug Reports

The Apple Developer Support group can address most of your scripting questions and process bug reports by sending email to: `devsupport@apple.com`.

Enhancements

We have a special email account so you can send your ideas directly to the Installer Team. Just mail us at: `installer@apple.com`. We are always looking for new ideas to improve Macintosh installation, and your suggestions will help us design installation technology to better address your needs.

Suggested Reading

We have assumed a certain level of knowledge of scriptwriters in the creation of this SDK. In addition to the documents provided with this SDK we suggest several documents that will help explain many of the terms and techniques used within this SDK.

- MPW: Macintosh Programmer's Workshop Reference. These volumes describes the gory details of the MPW environment in addition to reference sections on each of the standard tools we use in the examples (Make, Rez, DeRez). These manuals should have come with your MPW software. These manuals can also be found on the E.T.O.: Essentials•Tools•Objects CD. Contact APDA at `apda@apple.com` for purchasing information.
- TN-Platforms & Tools (note: PT35 - Stand-Alone Code, ad nauseam). This Tech. Note describes how to write stand-alone code resources. Since some of the advanced features of the Installer require the scriptwriter to create code resources that the Installer calls, this gives a good overview of what not to do. This document can be found in Adobe Acrobat PDF format on the Developer CD Reference Library edition.
- Inside Macintosh: Text. The chapter on the Font Manager gives an overview of font resources and how they are used. This might be handy for scriptwriters that are installing fonts for the first time and do not understand the relationship between the difference resource types and IDs. This document is available in book form from technical booksellers and in Adobe Acrobat PDF format on the Developer CD Reference Library edition.

SDK Contents

This SDK is split into two folders: Upgrader and its engines, and old Installer 4.0.8. Upgrader in conjunction with Installer Engine or ASR will replace the old Installer 4.0.X application as Apple's primary installation user experience. Most Installer scripts written for Installer 4.0.X can be used with Upgrader and Installer Engine, so switching should be easy for most of you.

Upgrader 1.2.3 & Engines

ASR 1.3.2

The Apple Software Restore application copies the contents of a disk image created by Disk Copy to the selected destination disk. Upgrader provides support for using ASR as an engine to install one or more software components from within an Upgrader-based program.

This folder contains documentation about creating a disk image for use with ASR, the ASR software, various helper AppleScripts, and the latest version of Disk Copy. See the document *Upgrader Guide* for information about using ASR with the installation plug-in.

DFA Server 1.0.2

This folder contains the latest version of DFA Server, which developers can use with the installation plug-in to check the destination disk prior to starting the installation. See the document *Upgrader Guide* for information about using DFA Server with the installation plug-in.

Installer Engine 4.5.2 UPDATED!

This folder contains the latest version of the Installer Engine application and associated files.

Installer Engine	A background-only application that executes Installer scripts. In combination with the Upgrader application, Installer Engine replaces the old Installer application. Since Installer Engine uses Apple events to communicate with a client application, developers and administrators can use Installer Engine to create various solutions.
Installer Engine Guide	Technical documentation for Installer Engine, including how to write Installer scripts and use the Installer Apple Event suite. (PDF)
Installer Debugger	An application for debugging Installer scripts and watching detailed progress of a running Installer Engine.
Installer Debugger Guide	A guide to using Installer Debugger. (PDF)
ObjectSupportLib	A PPC library implementing Apple event routines required by Installer Engine. Since Mac OS 8 has this library built in, you only need to include this file in the same folder as Installer Engine if wish to support running Installer Engine on pre-Mac OS 8 systems.

Installer Script Examples

User Interface Example

This Installer script provides an example of implementing a multiple-level feature hierarchy for custom install and multiple easy feature sets. The file copy uses the new ‘ifa#’ and ‘itf#’ resources to hold the file atoms and target specifications. This script is a good framework for creating you own Installer script that takes advantage of the user experience improvements in Upgrader 1.2.1.

Installer 4.0.8 Examples

An alias to the original example written for the Installer 4.0.X series of Installers. Since Installer Engine is backward compatible with most Installer scripts written for Installer 3.3, these examples are still valuable when writing Installer scripts for Installer Engine.

Interfaces

AIncludes

InstallerScript.a	Contains the necessary parameter blocks and callback definitions for creating code resources using 68K assembly.

CIncludes

ActionHandlerHeader.h	Contains a definition of the possible debugging IDs sent to a debugging client by Installer Engine.
InstallerAERegistry.h	Contains constants defining the Installer Apple event suite for use when writing a client application that will communicate with Installer Engine via Apple events.
InstallerScript.h	Contains the necessary parameter blocks and callback definitions for creating code resources using C or C++.
TargetInfoMgt.h	Contains the necessary parameter block, return result, and function definition for creating a ScriptCheck extension for use with an Atom Extender.
TargetVersMgt.h	Contains the necessary Contains the necessary parameter block, return result, and function definition for creating a ScriptCheck extension for use with a Version Compare code resource.

CSources

InstallerCallbackGlue.c	Contains the implementation of all callback routines supported by Installer Engine. This file can be included at the bottom of your C source file or compiled separately into a linkable library.

PIncludes

InstallerScript.p	Contains the necessary parameter blocks and callback definitions for creating code resources using Pascal.
TargetInfoMgt.p	Contains the necessary parameter block, return result, and function definition for creating a ScriptCheck extension for use with an Atom

Extender written in Pascal.

TargetVersMgt.p — Contains the necessary parameter block, return result, and function definition for creating a ScriptCheck extension for use with a Version Compare code resource written in Pascal.

RIncludes

InstallerTypes.r — Contains a Rez template for creating Installer scripts using the Rez language.

ScriptCheckTypes.r — Contains a Rez template for creating the 'insx' resource to be used by ScriptCheck when getting information about compressed files.

Templates

Installer 4.0.3-4.5 templates

Contains Installer resource templates for use with the utility application Resorcerer®. The Resorcerer application is an alternative to using ResEdit to view your rez'd Installer script. The provided templates will allow you to view most of the Installer resources in a recognizable format, where ResEdit displays many of these resources in binary format only. If you don't have a copy of Resorcerer, you may contact Mathemaesthetics, Inc. at: `sales@mathemaesthetics.com`.

Tools

Released

InstaCompOne 1.1.1

The InstaCompOne tools allow a scriptwriter to compress files and resources to save space on the source disk, then have Installer Engine automatically decompress the files and resources during installation.

FileAndRsrcSplitter Tool — An MPW tool for splitting files and resource items.

FONDEncoderTool — An MPW tool for including the 'FOND' resource item of a font in an InstaCompOne archive.

InstaCompOne 1.1.1 Guide — A guide for using InstaCompOne 1.1.1. (PDF)

InstaCompAtomExt.rsrc — The resources for including InstaCompOne compression/decompression atom extender in your Installer script.

InstaCompOneSCExt.rsrc — The resources for the ScriptCheck extension to be used with InstaCompOne.

InstaCompOneTool — An MPW tool for file and resource compression. This version of the tool includes a bug fix, adds additional information to the compressed archive header, and is compatible with previous version of InstaCompOne.

InstaCompOneTool 1.1.1 Notes

Provides further details on the bug fixes and updates to the InstaCompOneTool 1.1.

ScriptCheck 4.2b6

ScriptCheck	The ScriptCheck tool is used for checking the Installer script for errors and inconsistencies, and for automatically filling in certain fields such as file and resource item sizes, dates, etc. The ScriptCheck tool is run on an Installer script after the Installer script has been rez'd into resource form.
ScriptCheck Guide	A guide for using ScriptCheck. (PDF)

TomeViewer 1.3d3 UPDATED!

TomeViewer	Application for viewing and decompressing individual items contained in InstaCompOne compressed archives.
TomeViewer Read Me	Provides a brief description of TomeViewer and notes latest bug fixes and updates.

InstallerMPWTool 4.5.2

InstallerMPWTool	An MPW tool that can be used to automate Installer Engine.
InstallerMPWTool Guide	A guide to using InstallerMPWTool. (PDF)

Experimental

Cappella (Script Dev. Tool) UPDATED!

An installer for our visual script development tool. Check it out!

FileCrusher 1.3d3 UPDATED!

FileCrusher	An application which allows you to create, examine, and modify InstaCompOne compressed archive files (usually called "Tomes") for the Apple Installer, and assists in the creation of Installer scripts for Installer 4.0.8 or Installer Engine 4.5.X. FileCrusher does not create all the pieces necessary for an Installer script, but it will do most of the work for the first draft of a script and generate a source file that can be edited and run through Rez. We have estimated that Crusher can save 70% or more of the work needed to generate that first draft of a script. (And since the work saved is the least interesting part of the project, we estimate that Installer script developers will be 43% happier.)
Getting Started w/ Crusher	A guide for using FileCrusher 1.X. (PDF)
FileCrusher 1.3d3 Release Notes	Explains latest updates and bug fixes.

InstallTalk 1.0d1c1

Fat Example w/InstallTalk	An example that demonstrates use of the InstallTalk language.
InstallTalk Guide	A guide for using the InstallTalk language -- a

	language that allows creators of Apple Installer scripts to write their decision code in an AppleScript-like language, instead of the Installer's native rule language. Users familiar with HyperTalk, AppleScript, C, Pascal, BASIC or other high-level programming language will require very little effort to learn the InstallTalk language. (PDF)
InstallTalk	MPW tool to compile InstallTalk code into framework and rule resources, which can be included in an Installer script.

MachID Wannabe

MachID Wannabe	Control Panel that allows you to fake a machine's Gestalt machine type ('mach').

Upgrader 1.2.3 UPDATED!

Upgrader 1.2.3 App.

The Upgrader application provides a programming environment for creating assistant-like programs to guide users through the on-screen panels necessary to complete an installation or setup task. Each panel is a window that prompts the user to perform one step of the task. The panels are implemented by plug-ins, which are individual files containing code written by a developer that can be joined together into a single user experience. To control the order of the plug-ins and provide the information required by each plug-in, the developer creates a data file. The Upgrader application, plug-in files and data file are assembled by a developer into a group of files referred to as an Upgrader-based program. One example of a shipping Upgrader-based program is the Install Mac OS 8.5 program used to install Apple's Mac OS 8.5 system software.

Interfaces

CIncludes

InstallationPluginExt.h	A header file for writing a preflight extension code resource for the installation plug-in.
UpgraderPlugin.h	A header file for writing Upgrader plug-ins. It defines important constants and the routines supplied by the Upgrader shell.

RIncludes

UpgraderPluginTypes.r	Contains a Rez template for creating Upgrader data files using the Rez language.

Templates

Upgrader 1.2 Templates	Contains Upgrader data file resource templates for use with the utility application Resorcerer.

Libraries

PluginStubLib	A 68K Metrowerks library that provides access to the Upgrader shell routines. You'll need to link with this file in order to create an Upgrader plug-in.
DocViwerLib_68K	A 68K Metrowerks library that implements a

SimpleText document-compatible viewer window. Link with this file only if you need to view SimpleText documents outside of an Upgrader panel.

Extra Graphics

Additional pictures that you can use in your plug-in or data file.

ModifierTool 1.2.3a2 App.

ModifierTool allows modifications of an existing data file or creation of a new one by a developer or administrator so an Upgrader-based program can be tailored to the needs of an organization. See the document *Upgrader Guide* for information about how to use this tool.

Data File Examples

Simple ASR & Upgrader Example

An example of an Upgrader-based program that installs various files using both ASR and Installer Engine to perform the actual installation. This example can be used as a starting point for a developer or administrator that needs a similar installation solution.

Single Installer Script Example

An example of an Upgrader-based program that mimics the Installer 4.0.X application, providing a user experience that supports installation via a single Installer script. Since the Installer 4.0.X series will most likely not be developed further, this example provides any easy way developers to switch over to the new user interface.

Mac OS 8.5 Data File

An example of a complex data file to use a guide for your own data file.

Plug-in Examples

Simple App Launcher Plug-in

This example implements a single panel that launches an application with an optional document when the user clicks Continue. This might be helpful for a developer or administrator that wishes to have the user perform an essential step that cannot be fully integrated into the panel user experience. Examples of this might be running a third-party driver updating program or a virus scanning program that does not provide adequate Apple event support.

SAM Virus Checker Plug-in

This example implements a single panel that launches the SAM virus checker application when the user clicks Continue. The SAM application automatically starts scanning or repairing the disk. The user must manually quit the SAM application when it is done.

Preflight Function Example

Preflight functions are used in conjunction with the installation plug-in to determine if a software item should be hidden or checked at runtime, overriding the default attributes specified in the data file. This might be necessary if a specific software component installer is designed to work on a subset of the environments an Upgrader-based program supports.

Common Files

To conserve disk space, files used by all plug-in examples are stored here.

Additional Data File Tools

Count The Forks	An application used by the installation plug-in to calculate the installation size values to enter into the data file. See the document *Upgrader Guide* for information about calculating the installation size.
UpgraderCommandsMPWTool	An MPW tool that prints executable lines for driving Installer Engine using the InstallerMPWTool MPW Tool.
UpgraderCommandsTool Guide.pdf	A manual describing how to use the UpgraderCommandsMPWTool MPW Tool.
ReadDataFileLib.o	A 68K Metrowerks library that implements an API to the installation plug-in preference resource so another application could automate Installer Engine using information from an Upgrader data file.
ReadDataFileLib.h	A header file for the installation plug-in preference resource API.
ReadDataFileLib Guide.pdf	A manual describing how to use the installation plug-in preference resource API.

Network Assistant Helper Apps

Contains applications that are required to perform remote installations using Network Assistant. See the document *Upgrader Guide* for information about setting up an Upgrader-based program to be remotely installed using Network Assistant.

Installer 4.0.8

Installer

Version 4.0.8 of the Installer application. Installer 4.0.8 contains support for multi-CD installs, multilingual support, and bug fixes to Installer 4.0.6. See the Installer 4.0.8 Release Notes document for more details.

Installer 4.0.8 Technical Guide

A scripting guide for Installer 4.0.8. (PDF)

Installer Error Guide

A description of the potential errors presented by Installer 4.0.8. (PDF)

DeveloperInterfaces

AIncludes

InstallerScript.a	Contains the necessary parameter blocks and return result definitions for creating Rule Function, Action Atom, Version Compare Function, Target Search Function, Setup Function, and Atom

	Extender Function code resources using 68K assembly.
CIncludes	
ActionHandlerHeader.h	Contains the necessary parameter block, return result and function definition for creating Action Handlers.
InstallerScript.h	Contains the necessary parameter blocks, return result definitions, related callback routines and function definitions for creating format 0,1, and 2 Action Atom code resources, Atom Extender code resources, Rule Function code resources, File Search Procedure code resources, Setup Function code resources, and Version Compare code resources.
TargetInfoMgt.h	Contains the necessary parameter block, return result and function definition for creating a ScriptCheck extension for use with an Atom Extender.
TargetVersMgt.h	Contains the necessary Contains the necessary parameter block, return result and function definition for creating a ScriptCheck extension for use with a Version Compare code resource.
CSources	
InstallerCallbackGlue.c	Contains the implementation of the Action Atom, Action Handler, Atom Extender, and Installer Memory callback routines. This file can be included at the bottom of your C source files. Remember to include the files “ActionHandlerHeader.h” and "InstallerScript.h" at the top of your source file.
PIncludes	
InstallerScript.p	Contains the necessary parameter blocks, return result definitions, and function definitions for creating Action Atom, Action Handler, and File Search Procedure code resources using Pascal.
TargetInfoMgt.p	Contains the necessary Contains the necessary parameter block, return result and function definition for creating a ScriptCheck extension for use with an Atom Extender written in Pascal.
TargetVersMgt.p	Contains the necessary Contains the necessary Contains the necessary parameter block, return result and function definition for creating a ScriptCheck extension for use with a Version Compare code resource written in Pascal.
RIncludes	
InstallerTypes.r	Contains a Rez template for creating Installer scripts using the Rez tool language.
ScriptCheckTypes.r	Contains a Rez template for creating the ‘insx’ resource to be used by ScriptCheck when getting

information about compressed files.

Templates

Installer 4.0.3-4.0.8 templates

Contains Installer resource templates for use with the utility application Resorcerer®. The Resorcerer application is an alternative to using ResEdit to view your rez'd Installer script. The provided templates will allow you to view most of the Installer resources in a recognizable format, where ResEdit displays many of these resources in binary format only. If you don't have a copy of Resorcerer, you may contact Mathemaesthetics, Inc. at: `sales@mathemaesthetics.com`.

DisplayTEXT Startup Function

Demo DisplayTEXT Script	Demonstration Installer script.
Display TEXT Guide	Provides a guide for using the startup function. (PDF)
Display TEXT.r	'inpr' and related resources
Display TEXT.rsrc	The startup function code resource
SampleSplashScreen.rsrc	Sample Splash screen used in demo script.

Installer 4.0.8 Examples

Script Development FAQ

Contains frequently asked Installer scripting questions, and explains some gotchas.

Script Examples Guide

A guide to the example scripts. (PDF)

• Example Example

This example is intended as a guideline for working with the rest of the examples included with the Installer SDK. This example, using the InstaCompOne compression / decompression, installs a file to the target volume.

• Some Useful Utilities

Contains some useful MPW scripts for building the example scripts and an example Makefile that you can modify to use for your own Installer script projects.

Action Atoms [inaa] Example

This script shows how to implement the three versions of action atoms ('inaa') that are supported by the Installer (format0, format1, and format2).

Custom UI Example

This script demonstrates usage of the Global, Custom, and Easy Install frameworks. Also included are custom help pages, package comment ('inpc') resources, custom install subpackages, custom removal packages, installation of custom folder icons, and the method for allowing the user to select the volume and folder as the target for the installation. If you are using Installer Engine & Upgrader, please use UserInterfaceExample as a guide instead.

File Compare by Procedure

This script shows an example of using a custom version compare procedure to determine the version number of the target file.

File Compare Vers & Date

This script demonstrates comparison of files during installation based on either creation date or the file version.

File Compression/Split Example

This script demonstrates installation of files using InstaCompOne compression. Files are installed via compression, via splitting, and via splitting and compression.

Folder Merge [infm] Example

This script shows how to use the Folder Merge Atom ('infm') to install all the files within a folder.

Font Compression/Split Example

This script demonstrates installation of fonts using InstaCompOne compression. Fonts are installed via compression, via splitting, and via splitting and compression.

InstaCompOneExample

This script demonstrates installation of files and fonts using InstaCompOne compression. This example is detailed in the InstaCompOne Guide which serves as a tutorial on the use of InstaCompOne features.

Install App Fat/PPC/68K

This example gives an innovative (though complex) method for conserving disk space on the installation disks and providing the user with a simplified selection for installing an application from within Custom Install — 68K only, Power Macintosh only or both (Fat App).

PowerMac vs 68K Rules Example

This script is a highly simplified version of installing Power Macintosh or 68K software. Unlike the example above, it makes no effort to conserve disk space, or to provide a simplified Custom Install selection to the user. It is intended only for very simple projects, and gives a good example of how to determine whether the target platform is 68K or Power Macintosh.

Preference/Setup Func. Example

This script contains a preference resource ('inpr') which references a Setup Function code resource that searches the default target disk for the first folder that contains the SimpleText (or TeachText) application. If one is found, then it becomes the suggested target folder for the user.

ResList [inr#] Example

This script demonstrates usage of the ResList ('inr#') atom.

ResMerge [inrm] Example

This script demonstrates usage of the ResMerge ('inrm') atom.

Rsrc Compare by Procedure

This script shows an example of using a custom version compare procedure to

determine the version number of the target resource, by looking at the first long word of the target resource.

Rsrc Compression/Split Example

This script demonstrates installation of resource items using InstaCompOne compression. Resources are installed via compression, via splitting, and via splitting and compression.

Rule Function [inrf] Example

This script shows how to implement an Installer Rule Function. This example script uses a Rule Function to check the version of the currently running operating system.

Search Proc [insp] Example

This script shows an example of a File Search Procedure which uses PBCatSearch to find all copies of the SimpleText (or TeachText) application during an installation or removal. The File Search routine is called after the Install or remove button is clicked. The four usages of the search procedure demonstrated are : Replacing all found files, Removing all found files, Replacing a resource item in all found files, and Removing a resource item from all found files.

NOTE
All copies of the SimpleText (and TeachText) application will be updated or removed on the selected disk when you click the Install or Remove button. When running this script, take care not to target a HD onto which you have copied this SDK -- the script examples that depend on SimpleText will break.
♦

Simple Atom Extender

This example shows how to write a simple Atom Extender that copies a file directly. Also demonstrated is how to implement ScriptCheck extension resources.

System Rules

This script demonstrates usage of the Easy and Custom Install frameworks with rule clauses to determine the system version of the target volume. A different file is installed depending on which system version is found. If the system version is not supported then an error message is displayed to the user, and installation is prevented by disabling the Install button in the Installer dialog.

Installer Debugger 4.0.8

Installer 4.0.8 Debug	The Installer 4.0.8 application with debugging enabled.
Installer Debugger 4.0.3 Guide	A brief guide on the use of the Installer Debugger Action Handler. No changes where made between version 4.0.3 and 4.0.8. (PDF)
Installer Debugger.rsrc	The compiled resources for the Installer Debugger. Add these resources to your script to automatically enable the Installer Debugger 4.0.8.
Installer 4.0.8 Debug + rsrcs	A debug version of Installer 4.0.8 containing all the debug resources. With this installer, you will not need to copy the debug resources into your

installer script.

Localized Installers 4.0.8

Contains all the localized versions of Installer 4.0.8 available at this time.

More Help Pages

Examples of PICT’s to add to the help feature of the Installer, including the fonts used to create the standard help pages.