
Installer JavaScript Reference

Distribution Tools



2008-12-16



Apple Inc.
© 2008 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Carbon, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Numbers is a trademark of Apple Inc.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX is a registered trademark of The Open Group

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION,

EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction **Introduction to Installer JavaScript Reference 5**

JavaScript in Distribution Definition Files 5
Who Should Read this Document 5
Organization of this Document 5

Part I **Classes 7**

Chapter 1 **Applications 9**

Overview 9
Tasks 9
Methods 9

Chapter 2 **Choice 11**

Overview 11
Tasks 11
Properties 12

Chapter 3 **Files 15**

Overview 15
Tasks 15
Methods 15

Chapter 4 **IORegistry 17**

Overview 17
Tasks 17
Methods 17

Chapter 5 **ProcessInformation 21**

Overview 21

Chapter 6 **Result 23**

Overview 23
Tasks 23
Properties 23

Chapter 7 System 25

- Overview 25
- Tasks 25
- Properties 27
- Methods 28

Chapter 8 Target 33

- Overview 33
- Tasks 33
- Properties 33
- Methods 34

Document Revision History 35

Index 37

Introduction to Installer JavaScript Reference

Technology area:	Installer
Companion guide	Software Delivery Guide

JavaScript in Distribution Definition Files

A distribution definition file defines the install experience for a product. The Installer application opens and interprets distribution definition files to generate the user interface users manipulate to execute and customize a product install.

Distribution definition files contain XML and JavaScript code. The XML code defines the structure of a distribution, while the JavaScript code defines and manages install-time properties, such as installation options. *Distribution Definition Reference* describes the XML schema used to define the structure of a distribution. For details on distributions, see *Software Delivery Guide*.

This document describes the object model that the JavaScript code in a distribution definition file can use to manage user input and system properties.

Who Should Read this Document

This document is targeted mainly to creators of distribution packages.

Organization of this Document

This document contains chapters that describe each of the type of objects JavaScript code in distribution definition files can use. This document also contains a revision history.

INTRODUCTION

Introduction to Installer JavaScript Reference

Classes

Applications

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

The Applications object provides methods to obtain information about running applications. This object is accessed using the `system.applications` property.

Tasks

Getting Information About Running Applications

[fromPID](#) (page 10)

Provides information about a running application with a given process ID.

[fromIdentifier](#) (page 10)

Provides information about running processes with a given application identifier (bundle ID). For example, `com.apple.TextEdit`.

[all](#) (page 9)

Provides information about running applications registered with the process manager.

Methods

all

Provides information about running applications registered with the process manager.

`all`

Return Value

An array of dictionaries (associative arrays) describing the running applications registered with the process manager.

Availability

Available in Mac OS X v10.4 and later.

See Also[ProcessInformation](#) (page 21)**fromIdentifier**

Provides information about running processes with a given application identifier (bundle ID). For example, `com.apple.TextEdit`.

```
fromIdentifier(bundleID)
```

Parameters

bundleID

A string with the bundle ID of the desired application.

Return Value

An array of dictionaries (associative arrays) describing the running applications identified by *bundleID*.

Availability

Available in Mac OS X v10.4 and later.

See Also[ProcessInformation](#) (page 21)**fromPID**

Provides information about a running application with a given process ID.

```
fromPID(processID)
```

Parameters

processID

A string with the process ID of the desired application.

Return Value

A dictionary (associative array) describing the application identified by *processID*.

Availability

Available in Mac OS X v10.4 and later.

See Also[ProcessInformation](#) (page 21)

Choice

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

A Choice object represents a single installation choice. Choice objects are properties of the `choices` global variable and represent the installation choices defined in a distribution package.

The name of each property of the `choices` array corresponds to the `choice-id` attribute of each of the distribution's installation choices (see *Distribution Definition Reference* for more information). Therefore, a specific choice is accessed this way:

```
choices.<choice-id>
```

`<choice-id>` represents the identifier of the desired choice in the installation definition. For example, to access the `selected` property of a choice with `'choice_a'` as its identifier, you would use the following expression:

```
choices['choice_a'].selected
```

Tasks

Properties

- `selected` (page 13)
true when the choice is selected, false otherwise.
- `enabled` (page 12)
true when the choice is enabled, false otherwise.
- `visible` (page 13)
true when the choice is visible, false otherwise.
- `title` (page 13)
String with the title of the distribution package.
- `description` (page 12)
String with the description of the distribution.
- `packages` (page 12)

[packageUpgradeAction](#) (page 12)

Properties

description

String with the description of the distribution.

Availability

Available in Mac OS X v10.4 and later.

enabled

`true` when the choice is enabled, `false` otherwise.

Availability

Available in Mac OS X v10.4 and later.

packages

Discussion

Array identifying each package attached to the choice. Each item of the array references a `pkg-ref` element in the installation definition and contains the following two properties:

- `identifier`: The package identifier. Corresponds to the `id` attribute of the `pkg-ref` element.
- `version`: The package version. Corresponds to the `version` attribute of the `pkg-ref` element.

Availability

Available in Mac OS X v10.4 and later.

packageUpgradeAction

Discussion

String that specifies the relationship between the choice's packages (specified by the `packages` property) and the same packages (based on the package identifier) installed on the host. The possible values are:

- `'clean'`: None of the packages are installed.
- `'installed'`: All the packages are already installed.
- `'upgrade'`: The versions of all the packages are higher than the versions of the installed packages.
- `'downgrade'`: The versions of all the packages are lower than the versions of the installed packages.
- `'mixed'`: At least two of the following conditions are true:
 - One or more of the packages are upgrades.

Choice

- One or more of the packages are downgrades.
- One or more of the packages are not installed.

For example, if a distribution contains a choice with a required package, you may deselect that choice if the installed package is more recent.

Availability

Available in Mac OS X v10.4 and later.

selected

true when the choice is selected, false otherwise.

Availability

Available in Mac OS X v10.4 and later.

title

String with the title of the distribution package.

Availability

Available in Mac OS X v10.4 and later.

visible

true when the choice is visible, false otherwise.

Availability

Available in Mac OS X v10.4 and later.

Files

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

The Files object provides methods for accessing files. This object is accessed using the `system.files` property.

Tasks

Accessing Files

[fileExistsAtPath](#) (page 16)

Indicates whether the given path identifies an existent file.

[plistAtPath](#) (page 16)

Provides the contents of a given plist file.

[bundleAtPath](#) (page 15)

Provides the contents of the info plist file of the identified bundle.

Methods

bundleAtPath

Provides the contents of the info plist file of the identified bundle.

```
bundleAtPath(filePath)
```

Parameters

filePath

String with the path of the desired bundle.

Return Value

Dictionary (associative array) with the contents of the bundle's info plist file.

Availability

Available in Mac OS X v10.4 and later.

fileExistsAtPath

Indicates whether the given path identifies an existent file.

```
fileExistsAtPath(filePath)
```

Parameters

filePath

String with the path of the desired file.

Return Value

true when *filePath* identifies an existent file; false otherwise.

Availability

Available in Mac OS X v10.4 and later.

plistAtPath

Provides the contents of a given plist file.

```
plistAtPath(filePath)
```

Parameters

filePath

String with the path of the desired file.

Return Value

Dictionary (associative array) with the contents of the identified plist file.

Availability

Available in Mac OS X v10.4 and later.

IORegistry

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

The IORegistry object provides access to the IOKit registry. This object is accessed using the `system.ioregistry` property.

See *I/O Kit Fundamentals* for details about the IOKit registry.

Tasks

Accessing the IOKit Registry

- [fromPath](#) (page 18)
Provides a dictionary of the properties of a given IOKit object.
- [matchingClass](#) (page 18)
Provides the IOKit objects of a given class.
- [matchingName](#) (page 18)
Provides the IOKit objects of the specified name.
- [childrenOf](#) (page 17)
Provides the children of the given IOKit object.
- [parentsOf](#) (page 19)
Provides the parents of the given IOKit object.

Methods

childrenOf

Provides the children of the given IOKit object.

```
childrenOf(objectName)
```

Parameters

objectName

String with the path of the desired IOKit object or an item of an IOKit object array.

Return Value

Array of IOKit object dictionaries.

Availability

Available in Mac OS X v10.4 and later.

See Also

[fromPath](#) (page 18)

[matchingClass](#) (page 18)

[matchingName](#) (page 18)

fromPath

Provides a dictionary of the properties of a given IOKit object.

```
fromPath(filePath)
```

Parameters

filePath

String with the path of the desired IOKit object.

Return Value

Dictionary of the properties of the IOKit object.

Availability

Available in Mac OS X v10.4 and later.

matchingClass

Provides the IOKit objects of a given class.

```
matchingClass(className, [servicePlane])
```

Parameters

className

String with an IOKit class name.

servicePlane

Optional. String with the service plane to search. When unspecified, 'IOServicePlane' is used.

Return Value

Array of IOKit object dictionaries.

Availability

Available in Mac OS X v10.4 and later.

matchingName

Provides the IOKit objects of the specified name.

`matchingName(objectName, [servicePlane])`

Parameters

objectName

String with an IOKit object name.

servicePlane

Optional. String with the service plane to search. When unspecified, 'IOServicePlane' is used.

Return Value

Array of IOKit object dictionaries.

Availability

Available in Mac OS X v10.4 and later.

parentsOf

Provides the parents of the given IOKit object.

`childrenOf(objectName)`

Parameters

objectName

String with the path of the desired IOKit object or an item of an IOKit object array.

Return Value

Array of IOKit object dictionaries.

Availability

Available in Mac OS X v10.4 and later.

See Also

[fromPath](#) (page 18)

[matchingClass](#) (page 18)

[matchingName](#) (page 18)

ProcessInformation

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

A `ProcessInformation` object is a dictionary (associative array) describing an application.

The following table describes the items available in the dictionary.

Key	Description
'PSN'	The process serial number. See <code>ProcessSerialNumber</code> .
'Parent PSN'	The process serial number of the application that launched the application in question.
'FileType'	The file type (if any) of the executable.
'FileCreator'	The creator type (if any) of the executable.
'BundlePath'	The path to the application bundle (if the application is bundled).
'pid'	The UNIX PID for the program in question.
'LSBackgroundOnly'	true when the application is a background-only application; false otherwise.
'LSUIElement'	true when the application is an accessibility UIElement; false otherwise.
'IsHiddenAttr'	true if the application is currently hidden; false otherwise.
'RequiresClassic'	true if the application's Info.plist file indicates that it requires Classic; false otherwise.
'RequiresCarbon'	true if the application's Info.plist file indicates that it is a Carbon application; false otherwise.

See the discussion in `ProcessInformationCopyDictionary` for additional information.

Result

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

The Result object provides methods to obtain information about the result of an Installation Check or Volume Check script. This object is accessed using the `my.result` property.

Tasks

Properties

[message](#) (page 23)

[type](#) (page 24)

[title](#) (page 23)

Properties

message

Discussion

String describing the cause of a warning or fatal result.

title

Discussion

The title of the warning.

type

Discussion

String with the type of the result. Its values can be:

- 'Warn': The user is notified that a requirement is not met but can continue the install.
- 'Fatal': For Installation Check scripts, the user cannot continue the install. For Volume Check scripts, the user cannot install on the affected volume.
-

System

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

The System object provides access to information about the target host. This object is accessed using the `system` global variable.

Tasks

Properties

[applications](#) (page 27)

[defaults](#) (page 27)

[files](#) (page 27)

[ioregistry](#) (page 27)

[users.desktopSessionCount](#) (page 27)

[version](#) (page 28)

Accessing Object Properties

[propertiesOf](#) (page 30)

Provides an array with the properties of a given object.

Accessing System Information

`gestalt` (page 28)

Provides gestalt information that corresponds to the given selector.

`sysctl` (page 32)

Provides the result of a `sysctlbyname()` call using the given selector.

Comparing Version Numbers

`compareVersions` (page 28)

Provides the result of comparing two given version strings (for example, '10.3.1' and '10.4').

Internationalizing Distributions

`LocalizedString` (page 29)

Provides the localized string in the installation package for the current locale for a given key.

`LocalizedStringWithFormat` (page 30)

Provides the formatted localized string in the installation package for the current locale for a given key and a given set of additional arguments.

`LocalizedStringStandardString` (page 29)

Provides the localized standard string in the installation package for the current locale for a given key.

`LocalizedStringStandardStringWithFormat` (page 29)

Provides the formatted localized string in the installation package for the current locale for a given key and a given set of additional arguments.

Logging

`log` (page 30)

Generates an Installer log entry with the `JS:` prefix.

Running Programs

`run` (page 31)

Launches a given program in the `Resources` directory of the installation package.

`runOnce` (page 31)

Launches a given program in the `Resources` directory of the installation package. On subsequent invocations with the same arguments, the program is not launched and the exit code of the first execution is returned.

Properties

applications

Discussion

Provides methods to obtain information about running applications. See [Applications](#) (page 9).

Availability

Available in Mac OS X v10.4 and later.

defaults

Discussion

Dictionary representing the defaults system.

Availability

Available in Mac OS X v10.4 and later.

files

Discussion

Provides file-access methods. See [Files](#) (page 15).

Availability

Available in Mac OS X v10.4 and later.

ioregistry

Discussion

Provides access to the IOKit registry. See [IORegistry](#) (page 17).

Availability

Available in Mac OS X v10.4 and later.

users.desktopSessionCount

Discussion

Number of logged-in users. (Includes only those users with a Window Server connection, that is, those that appear in the Fast User Switching list.)

Availability

Available in Mac OS X v10.4 and later.

version

Discussion

Dictionary (associative array) with the contents of `/System/Library/CoreServices/SystemVersion.plist`. See `sw_vers(1)`.

Availability

Available in Mac OS X v10.4 and later.

Methods

compareVersions

Provides the result of comparing two given version strings (for example, '10.3.1' and '10.4').

```
compareVersions(versionA, versionB)
```

Parameters

versionA

String with the first operand version.

versionB

String with the second operand version.

Return Value

-1 when *versionA* is lower than *versionB*, 0 when *versionA* is equal to *versionB*, and 1 when *versionA* is higher than *versionB*.

Availability

Available in Mac OS X v10.4 and later.

gestalt

Provides gestalt information that corresponds to the given selector.

```
gestalt(selector)
```

Parameters

selector

String specifying the selector for the data to retrieve.

Return Value

Data for the given selector.

Discussion

For example, this expression returns '1040' on Mac OS X v10.4.0:

```
system.gestalt('sysv').toString(16)
```

See *Gestalt Manager Reference* for detailed information.

Availability

Available in Mac OS X v10.4 and later.

localizedStandardString

Provides the localized standard string in the installation package for the current locale for a given key.

```
localizedStandardString(stringKey)
```

Parameters

stringKey

String that identifies the desired localized string.

Return Value

Localized string, if found in the installation package; `null` otherwise.

Availability

Available in Mac OS X v10.4 and later.

localizedStandardStringWithFormat

Provides the formatted localized string in the installation package for the current locale for a given key and a given set of additional arguments.

```
localizedStandardStringWithFormat(stringKey, args...)
```

Parameters

stringKey

String that identifies the desired localized string.

args...

Arguments that replace placeholders (`%@`) in the formatted localized string.

Return Value

Localized string, if found in the installation package; `null` otherwise.

Availability

Available in Mac OS X v10.4 and later.

localizedString

Provides the localized string in the installation package for the current locale for a given key.

```
localizedString(stringKey)
```

Parameters

stringKey

String that identifies the desired localized string.

Return Value

Localized string, if found in the installation package; `null` otherwise.

Availability

Available in Mac OS X v10.4 and later.

localizedStringWithFormat

Provides the formatted localized string in the installation package for the current locale for a given key and a given set of additional arguments.

```
localizedStringWithFormat(stringKey, args...)
```

Parameters

stringKey

String that identifies the desired localized string.

args...

Arguments that replace placeholders (%@) in the formatted localized string.

Return Value

Localized string, if found in the installation package; `null` otherwise.

Availability

Available in Mac OS X v10.4 and later.

log

Generates an Installer log entry with the JS: prefix.

```
log(logString)
```

Parameters

logString

Log-entry text.

Discussion

For some objects, this function may print data that is unavailable or formatted differently when accessed in JavaScript code. To ensure that only data accessible in JavaScript code is shown in the log entry, invoke this function this way:

```
system.log(foo + '')
```

Such invocation forces the JavaScript interpreter to evaluate the expression and pass a String object to the function.

Availability

Available in Mac OS X v10.4 and later.

propertiesOf

Provides an array with the properties of a given object.

```
propertiesOf(object)
```

Parameters

object

An object.

Return Value

Array with the properties of *object*.

Availability

Available in Mac OS X v10.4 and later.

run

Launches a given program in the `Resources` directory of the installation package.

```
run(programName, args...)
```

Parameters

programName

Name of the program to execute.

args...

Arguments passed to the program.

Return Value

After the program is executed, its exit code. If the program is not executed (see discussion), `undefined`.

Discussion

The `allow-external-scripts` attribute of the `options` element in the distribution definition must be 'yes' for the program to be launched. And, the Installer application asks the user's permission before launching the program. See *Distribution Definition Reference* for more information.

Availability

Available in Mac OS X v10.4 and later.

See Also

[runOnce](#) (page 31)

runOnce

Launches a given program in the `Resources` directory of the installation package. On subsequent invocations with the same arguments, the program is not launched and the exit code of the first execution is returned.

```
run(programName, args...)
```

Parameters

programName

Name of the program to execute.

args...

Arguments passed to the program.

Return Value

After the program is executed, its exit code. If the program is not executed (see discussion), `undefined`.

Discussion

The `allow-external-scripts` attribute of the `options` element in the distribution definition must be 'yes' for the program to be launched. And, the Installer application asks the user's permission before launching the program. See *Distribution Definition Reference* for more information.

Use this method instead of [run](#) (page 31) when a program's exit code using the same arguments is not expected to change.

Availability

Available in Mac OS X v10.4 and later.

sysctl

Provides the result of a `sysctlbyname()` call using the given selector.

```
sysctl(selector)
```

Parameters

selector

String with the hardware selector for the desired data.

Return Value

Hardware data corresponding to *selector*.

Discussion

See `/usr/include/sys/sysctl.h` for the list of hardware selectors.

Availability

Available in Mac OS X v10.4 and later.

Target

Technology area:	Installer
Availability	Available in Mac OS X v10.4 and later.

Overview

The Target object provides methods to obtain information about the installation volume. This object is accessed using the `my.target` property.

Tasks

Properties

[mountpoint](#) (page 34)

[availableKilobytes](#) (page 33)

[systemVersion](#) (page 34)

Accessing Installation Receipts

[receiptForIdentifier](#) (page 34)

Provides information about the receipt for a given package on the installation host.

Properties

availableKilobytes

Discussion

Available space in the installation volume, in kilobytes.

Target

mountpoint

Discussion

String with that identifies the installation volume's mountpoint.

systemVersion

Discussion

String specifying the version of the host's operating system.

Methods

receiptForIdentifier

Provides information about the receipt for a given package on the installation host.

```
receiptForIdentifier(packageID)
```

Parameters

packageID

String specifying a package identifier. For example, `com.apple.TextEdit.pkg`.

Return Value

Dictionary with information about the receipt for the specified package.

Availability

Available in Mac OS X v10.4 and later.

Document Revision History

This table describes the changes to *Installer JavaScript Reference*.

Date	Notes
2008-12-16	New document that describes the object model available to JavaScript code in distribution definition files.

REVISION HISTORY

Document Revision History

Index

A

all [method 9](#)
applications [property 27](#)
availableKilobytes [property 33](#)

B

bundleAtPath [method 15](#)

C

childrenOf [method 17](#)
compareVersions [method 28](#)

D

defaults [property 27](#)
description [property 12](#)

E

enabled [property 12](#)

F

fileExistsAtPath [method 16](#)
files [property 27](#)
fromIdentifier [method 10](#)
fromPath [method 18](#)
fromPID [method 10](#)

G

gestalt [method 28](#)

I

ioregistry [property 27](#)

L

localizedStandardString [method 29](#)
localizedStandardStringWithFormat [method 29](#)
localizedString [method 29](#)
localizedStringWithFormat [method 30](#)
log [method 30](#)

M

matchingClass [method 18](#)
matchingName [method 18](#)
message [property 23](#)
mountpoint [property 34](#)

P

packages [property 12](#)
packageUpgradeAction [property 12](#)
parentsOf [method 19](#)
plistAtPath [method 16](#)
propertiesOf [method 30](#)

R

receiptForIdentifier [method 34](#)

run [method 31](#)
runOnce [method 31](#)

S

selected [property 13](#)
sysctl [method 32](#)
systemVersion [property 34](#)

T

title [property 13, 23](#)
type [property 24](#)

U

users.desktopSessionCount [property 27](#)

V

version [property 28](#)
visible [property 13](#)