Automator Reference Update

Apple Applications > Automator

2007-07-25

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

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 to Automator Reference Update 5

Organization of This Document 5
See Also 5

10.5 Symbol Changes 7

Classes 7
AMAction 7
AMWorkflow (New) 7
AMWorkflowController (New) 8
AMWorkflowView (New) 9
C Symbols 9
AMBundleAction.h 10
AMWorkflowController.h 10
AutomatorErrors.h 10

10.4 Symbol Changes 13

Classes 13
AMAction (New) 13
AMAppleScriptAction (New) 14
AMBundleAction (New) 14
AMShellScriptAction (New) 14

Document Revision History 15

Introduction to Automator Reference Update

This document summarizes the symbols that have been added to the Automator framework. The full reference documentation notes in what version a symbol was introduced, but sometimes it's useful to see only the new symbols for a given release.

If you are not familiar with this framework you should refer to the complete framework reference documentation.

Organization of This Document

Symbols are grouped by class or protocol for Objective-C and by header file for C. For each symbol there is a link to complete documentation, if available, and a brief description, if available.

See Also

For reference documentation on this framework, see *Automator Framework Reference*.

10.5 Symbol Changes

This article lists the symbols added to `Automator.framework` in Mac OS X v10.5.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

AMAction

Complete reference information is available in the `AMAction` reference.

Instance Methods

`closed`	Invoked by Automator when the receiving action is removed from a workflow, allowing it to perform cleanup operations.
`didFinishRunningWithError:`	Sent by the receiver to itself when it has finished running asynchronously.
`ignoresInput`	Returns a Boolean value that indicates whether the action acts upon its input or the input is ignored.
`initWithContentsOfURL:error:`	Loads an Automator action from a file URL.
`name`	Returns the name of the action.
`output`	Returns the receiver's output.
`runAsynchronouslyWithInput:`	Causes Automator to wait for notification that the receiver has completed execution, which allows the receiver to perform an asynchronous operation.
`willFinishRunning`	Invoked by Automator when the receiver has essentially completed its run phase.

AMWorkflow (New)

Complete reference information is available in the `AMWorkflow` reference.

Class Methods

`runWorkflowAtURL:withInput:error:`	Loads and runs the specified workflow file.

Instance Methods

`actions`	Returns an array of the workflow’s actions.
`addAction:`	Adds the specified action at the end of the receiving workflow.
`fileURL`	Returns a URL that specifies the location of the workflow file.
`initWithContentsOfURL:error:`	Creates and initializes a workflow based on the contents of the specified file.
`input`	Returns the input data that is passed to the first action in the workflow.
`insertAction:atIndex:`	Inserts the specified action at the specified position of the receiving workflow.
`moveActionAtIndex:toIndex:`	Moves the action from the specified start position to the specified end position in the receiving workflow.
`removeAction:`	Removes the specified action from the receiver.
`setInput:`	Sets the input data that is passed to the first action in the workflow.
`setValue:forVariableWithName:`	Sets the value of the workflow variable with the specified name.
`valueForVariableWithName:`	Returns the value of the workflow variable with the specified name.
`writeToURL:error:`	Writes the workflow to the specified file.

AMWorkflowController (New)

Complete reference information is available in the `AMWorkflowController` reference.

Instance Methods

`canRun`	Returns a Boolean value that indicates whether the receiver's workflow is able to run.
`delegate`	Returns the receiver's delegate.
`isRunning`	Returns a Boolean value that indicates whether the receiver's workflow is currently running.
`run:`	Runs the associated workflow, after first clearing any results stored by its actions during any previous run.

setDelegate:	Sets the receiver's delegate.
setWorkflow:	Sets the receiver's workflow.
setWorkflowView:	Sets the receiver's workflow view.
stop:	Stops the associated workflow.
workflow	Returns the receiver's workflow.
workflowView	Returns the receiver's workflow view.

Delegate Methods

workflowController:didError:	Invoked when the receiver's workflow encounters an error.
workflowController:didRunAction:	Invoked when an action in the receiver's workflow is finished running.
workflowController:willRunAction:	Invoked when an action in the receiver's workflow is about to run.
workflowControllerDidRun:	Invoked after the workflow is run.
workflowControllerDidStop:	Invoked after the workflow stops.
workflowControllerWillRun:	Invoked when the workflow is about to run.
workflowControllerWillStop:	Invoked when the workflow is about to stop.

AMWorkflowView (New)

Complete reference information is available in the AMWorkflowView reference.

Instance Methods

isEditable	Returns a Boolean value indicating whether the workflow view is editable.
setEditable:	Sets whether the workflow view is editable.

C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

AMBundleAction.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

IBOutlet	Identifier used to qualify an instance-variable declaration so that Interface Builder can synchronize the display and connection of outlets with Xcode.

AMWorkflowController.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

workflowView	

AutomatorErrors.h

Data Types & Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AMActionApplicationResourceError	An application required by the action is not found.
AMActionApplicationVersionResourceError	An application required by the action is the wrong version.
AMActionArchitectureMismatchError	The action's binary is not compatible with the current processor; actions compiled for PowerPC, for example, would encounter this error on Intel systems.
AMActionErrorKey	Use this key to obtain, from the userInfo dictionary of an instance of NSError, a reference to the action (class AMAction) that caused the error.
AMActionExceptionError	An action encounters an exception while running.
AMActionExecutionError	An action encounters an error while running.
AMActionFileResourceError	A file required by the action is not found.

`AMActionInitializationError`	Automator is unable to initialize an action (reason unknown).
`AMActionInsufficientDataError`	The action requires input data to run, but none was supplied.
`AMActionIsDeprecatedError`	The action has been deprecated. Use a replacement action, if available.
`AMActionLicenseResourceError`	A license required by the action was not found. The only license currently supported is QuickTime Pro.
`AMActionLinkError`	The action's executable failed to load due to linking issues.
`AMActionLoadError`	The action's executable failed to load; for example, there may have been a problem with a library it depends on.
`AMActionNotLoadableError`	The action's executable is of a type that is not loadable in the current process. If the action uses a custom subclass of AMBundleAction or AMAppleScriptAction, then the most likely problem is that the bundle's executable is missing or the NSPrincipleClass is not set in the Info.plist. Users are likely to be confused by a “missing bundle” error, so Automator presents it as the more generic “action not loadable” error.
`AMActionPropertyListInvalidError`	The property list (Info.plist) for an action is invalid; it could be missing, damaged, or constructed improperly.
`AMActionRequiredActionResourceError`	An action required by the action is not loaded.
`AMActionRuntimeMismatchError`	An attempt was made to load an action that is not compiled in a way that is compatible with the current application; for example, the action may be compiled for 32-bit applications or it may not be compatible with garbage collection.
`AMAutomatorErrorDomain`	This constant defines the Automator error domain.
`AMConversionFailedError`	Occurs when, for example, the converter encounters an error converting data from one type to another.
`AMConversionNoDataError`	The converter determines that a given conversion, though possible, would produce a nil result. Not displayed to the user.
`AMConversionNotPossibleError`	The converter determines that it is unable to convert from one data type to another. Not displayed to the user.

AMNoSuchActionError	The action could not be located on the system.
AMUserCanceledError	The user cancelled. This error is the same as the AppleScript error userCanceledErr, defined in MacErrors.h. When an Apple Event is canceled by the user, a running action may return this error. Automator ignores the error and stops the workflow gracefully, without displaying the error to the user.
AMWorkflowNewerActionVersionError	An action in a workflow is newer than the installed action; this error is presented to the user as a warning.
AMWorkflowNewerVersionError	Attempt to open a workflow document that was saved with a newer version of Automator.
AMWorkflowOlderActionVersionError	An action in a workflow is older than the installed action; this error is presented to the user as a warning.
AMWorkflowPropertyListInvalidError	Attempt to open a workflow document whose property list (document.wflow) could not be read; the property list document could be missing, damaged, or constructed improperly.

10.4 Symbol Changes

This article lists the symbols added to `Automator.framework` in Mac OS X v10.4.

Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

AMAction (New)

Complete reference information is available in the `AMAction` reference.

Instance Methods

`activated`	Invoked when the window of the Automator workflow to which the receiver belongs becomes the main window. This allows the receiver to synchronize its information with settings in another application.
`definition`	Returns the definition of the receiver.
`initWithDefinition:fromArchive:`	Initializes the receiver with the specified definition.
`opened`	Invoked when the receiver is first added to a workflow, allowing it to initialize its user interface.
`parametersUpdated`	Requests the receiver to update its user interface from its stored parameters, which have changed.
`reset`	Resets the receiver to its initial state.
`runWithInput:fromAction:error:`	Requests the receiver to perform its task using the specified input from the specified action.
`stop`	Stops the receiver from running.
`updateParameters`	Requests the receiver to update its stored set of parameters from the settings in the action's user interface.
`writeToDictionary:`	Examines the parameters and other configuration information specified in the passed dictionary and add its own information to it if appropriate.

AMAppleScriptAction (New)

Complete reference information is available in the `AMAppleScriptAction` reference.

Instance Methods

`script`	Returns the OSAScript object representing the receiver's script containing the on run command handler.
`setScript:`	Set's the receiver's script to newScript.

AMBundleAction (New)

Complete reference information is available in the `AMBundleAction` reference.

Instance Methods

`awakeFromBundle`	Sent to the receiver when all objects in its bundle have been unarchived.
`bundle`	Returns the receiver's bundle object.
`hasView`	Returns whether the receiver has a view associated with it.
`initWithDefinition:fromArchive:`	Initializes and returns an allocated AMBundleAction object.
`parameters`	Returns the receiver's parameters.
`setParameters:`	Sets the parameters of the receiver to newParameters.
`view`	Returns the receiver's view object.

AMShellScriptAction (New)

Complete reference information is available in the `AMShellScriptAction` reference.

Instance Methods

`inputFieldSeparator`	Returns the character that is used as the delimiter between items in the string passed in through standard input.
`outputFieldSeparator`	Returns the character used as a delimiter in the string output of the receiver.
`remapLineEndings`	Returns whether you want automatic remapping of carriage return (\r) to newline (\n) characters in the input string.

Document Revision History

This table describes the changes to *Automator Reference Update*.

Date	Notes
2007-07-25	Updated for Automator constants and new classes in Mac OS X version 10.5; the new classes are AMWorkflow, AMWorkflowController, and AMWorkflowView.
2005-06-06	Updated to reflect the new AMShellScriptAction class.
2005-04-29	New document that summarizes the symbols added to the Automator framework in Mac OS X v10.4.