
AMWorkflowController Class Reference

[Apple Applications](#) > [Automator](#)



2007-07-09



Apple Inc.
© 2007 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, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

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

AMWorkflowController Class Reference 5

Overview	5
Tasks	5
Accessing the Workflow	5
Accessing the Workflow View	6
Accessing the Delegate	6
Controlling the Workflow	6
Running an Action in the Workflow	6
Errors Messages	7
Instance Methods	7
canRun	7
delegate	7
isRunning	8
run:	8
setDelegate:	8
setWorkflow:	9
setWorkflowView:	9
stop:	9
workflow	10
workflowView	10
Delegate Methods	10
workflowController:didError:	10
workflowController:didRunAction:	11
workflowController:willRunAction:	11
workflowControllerDidRun:	12
workflowControllerDidStop:	12
workflowControllerWillRun:	12
workflowControllerWillStop:	12

Document Revision History 15

Index 17

AMWorkflowController Class Reference

Inherits from	NSController : NSObject
Conforms to	NSCoding (NSController) NSObject (NSObject)
Framework	/System/Library/Frameworks/Automator.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	AMWorkflowController.h
Companion guides	Automator Programming Guide Automator Framework Reference
Related sample code	AutoSample

Overview

The `AMWorkflowController` class lets you manage an Automator workflow in your application. You use the `AMWorkflow` class to instantiate a workflow and an instance of `AMWorkflowView` to display it.

A controller can run and stop a workflow and obtain information about its state. The controller's delegate can receive messages as the workflow is executed and its actions are run.

You can load and run a workflow with minimal overhead by using the `AMWorkflow` class method `runWorkflowAtURL:withInput:error:`. When you use `AMWorkflowController`, you get more control of the process, but there's more work, as you must create and initialize both the workflow and the workflow controller objects.

A workflow is run in a separate process so that any actions it contains are executed in a separate memory space. That helps to insulate your application from crashes, memory leaks, or exceptions that might occur from running the actions in a workflow.

Tasks

Accessing the Workflow

- `setWorkflow:` (page 9)
Sets the receiver's workflow.

- [workflow](#) (page 10)
Returns the receiver's workflow.

Accessing the Workflow View

- [setWorkflowView:](#) (page 9)
Sets the receiver's workflow view.
- [workflowView](#) (page 10)
Returns the receiver's workflow view.

Accessing the Delegate

- [delegate](#) (page 7)
Returns the receiver's delegate.
- [setDelegate:](#) (page 8)
Sets the receiver's delegate.

Controlling the Workflow

- [canRun](#) (page 7)
Returns a Boolean value that indicates whether the receiver's workflow is able to run.
- [isRunning](#) (page 8)
Returns a Boolean value that indicates whether the receiver's workflow is currently running.
- [run:](#) (page 8)
Runs the associated workflow, after first clearing any results stored by its actions during any previous run.
- [stop:](#) (page 9)
Stops the associated workflow.
- [workflowControllerWillRun:](#) (page 12) *delegate method*
Invoked when the workflow is about to run.
- [workflowControllerDidRun:](#) (page 12) *delegate method*
Invoked after the workflow is run.
- [workflowControllerWillStop:](#) (page 12) *delegate method*
Invoked when the workflow is about to stop.
- [workflowControllerDidStop:](#) (page 12) *delegate method*
Invoked after the workflow stops.

Running an Action in the Workflow

- [workflowController:didRunAction:](#) (page 11) *delegate method*
Invoked when an action in the receiver's workflow is finished running.
- [workflowController:willRunAction:](#) (page 11) *delegate method*
Invoked when an action in the receiver's workflow is about to run.

Errors Messages

- [workflowController:didError:](#) (page 10) *delegate method*
Invoked when the receiver's workflow encounters an error.

Instance Methods

canRun

Returns a Boolean value that indicates whether the receiver's workflow is able to run.

- (BOOL)canRun

Return Value

YES if the controller's workflow is able to run; NO otherwise.

Discussion

You might use this method to determine when to enable a "Run" button or other UI element you use to run the workflow.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [isRunning](#) (page 8)

Declared In

AMWorkflowController.h

delegate

Returns the receiver's delegate.

- (id)delegate

Return Value

The controller's delegate.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setDelegate:](#) (page 8)

Declared In

AMWorkflowController.h

isRunning

Returns a Boolean value that indicates whether the receiver's workflow is currently running.

- (BOOL)isRunning

Return Value

YES if the controller's workflow is currently running; NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [canRun](#) (page 7)

Declared In

AMWorkflowController.h

run:

Runs the associated workflow, after first clearing any results stored by its actions during any previous run.

- (IBAction)run:(id)sender

Parameters

sender

Object that initiated the run action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [stop:](#) (page 9)

Declared In

AMWorkflowController.h

setDelegate:

Sets the receiver's delegate.

- (void)setDelegate:(id)delegate

Parameters

delegate

The delegate object to set. This object will receive updates on the progress and state of the workflow controller.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [delegate](#) (page 7)

Declared In

AMWorkflowController.h

setWorkflow:

Sets the receiver's workflow.

- (void)setWorkflow:(AMWorkflow *)*workflow*

Parameters

workflow

A workflow object.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [workflow](#) (page 10)

Declared In

AMWorkflowController.h

setWorkflowView:

Sets the receiver's workflow view.

- (void)setWorkflowView:(AMWorkflowView *)*view*

Parameters

view

A workflow view object.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [workflowView](#) (page 10)

Declared In

AMWorkflowController.h

stop:

Stops the associated workflow.

- (IBAction)stop:(id)*sender*

Parameters

sender

Object that initiated the stop action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [run:](#) (page 8)

Declared In

AMWorkflowController.h

workflow

Returns the receiver's workflow.

- (AMWorkflow *)workflow

Return Value

The controller's workflow.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setWorkflow:](#) (page 9)

Declared In

AMWorkflowController.h

workflowView

Returns the receiver's workflow view.

- (AMWorkflowView *)workflowView

Return Value

The controller's workflow view.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setWorkflowView:](#) (page 9)

Declared In

AMWorkflowController.h

Delegate Methods

workflowController:didError:

Invoked when the receiver's workflow encounters an error.

```
- (void)workflowController:(AMWorkflowController *)controller didError:(NSError *)error
```

Parameters

controller

The controller object sending the message.

error

If a workflow error occurs, upon return contains an instance of `NSError` that describes the problem.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`AMWorkflowController.h`

workflowController:didRunAction:

Invoked when an action in the receiver's workflow is finished running.

```
- (void)workflowController:(AMWorkflowController *)controller didRunAction:(AMAction *)action
```

Parameters

controller

The controller object sending the message.

action

The workflow action that ran.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`AMWorkflowController.h`

workflowController:willRunAction:

Invoked when an action in the receiver's workflow is about to run.

```
- (void)workflowController:(AMWorkflowController *)controller willRunAction:(AMAction *)action
```

Parameters

controller

The controller object sending the message.

action

The workflow action that will run.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`AMWorkflowController.h`

workflowControllerDidRun:

Invoked after the workflow is run.

```
- (void)workflowControllerDidRun:(AMWorkflowController *)controller
```

Parameters

controller

The controller object sending the message.

Availability

Available in Mac OS X v10.5 and later.

Declared In

AMWorkflowController.h

workflowControllerDidStop:

Invoked after the workflow stops.

```
- (void)workflowControllerDidStop:(AMWorkflowController *)controller
```

Parameters

controller

The controller object sending the message.

Availability

Available in Mac OS X v10.5 and later.

Declared In

AMWorkflowController.h

workflowControllerWillRun:

Invoked when the workflow is about to run.

```
- (void)workflowControllerWillRun:(AMWorkflowController *)controller
```

Parameters

controller

The controller object sending the message.

Availability

Available in Mac OS X v10.5 and later.

Declared In

AMWorkflowController.h

workflowControllerWillStop:

Invoked when the workflow is about to stop.

```
- (void)workflowControllerWillStop:(AMWorkflowController *)controller
```

Parameters

controller

The controller object sending the message.

Availability

Available in Mac OS X v10.5 and later.

Declared In

AMWorkflowController.h

Document Revision History

This table describes the changes to *AMWorkflowController Class Reference*.

Date	Notes
2007-07-09	New document that describes the class used to manage an Automator workflow in an application.

REVISION HISTORY

Document Revision History

Index

C

canRun [instance method 7](#)

D

delegate [instance method 7](#)

I

isRunning [instance method 8](#)

R

run: [instance method 8](#)

S

setDelegate: [instance method 8](#)
setWorkflow: [instance method 9](#)
setWorkflowView: [instance method 9](#)
stop: [instance method 9](#)

W

workflow [instance method 10](#)
workflowController:didError: <NSObject> delegate
method [10](#)
workflowController:didRunAction: <NSObject>
delegate method [11](#)
workflowController:willRunAction: <NSObject>
delegate method [11](#)

workflowControllerDidRun: <NSObject> delegate
method [12](#)
workflowControllerDidStop: <NSObject> delegate
method [12](#)
workflowControllerWillRun: <NSObject> delegate
method [12](#)
workflowControllerWillStop: <NSObject> delegate
method [12](#)
workflowView [instance method 10](#)