

---

# AMWorkflow Class Reference

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



2006-10-26



Apple Inc.  
© 2006 Apple Computer, 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, AppleScript, Mac, Mac OS, and Xcode 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

---

## **AMWorkflow Class Reference 5**

---

Overview	5
Tasks	6
Running a Specified Workflow	6
Creating and Initializing a Workflow	6
Saving Changes to a Workflow	6
Getting Information About the Workflow	6
Manipulating the Workflow's Actions	6
Class Methods	7
runWorkflowAtURL:withInput:error:	7
Instance Methods	7
actions	7
addAction:	8
fileURL	8
initWithContentsOfURL:error:	8
input	9
insertAction:atIndex:	9
moveActionAtIndex:toIndex:	10
removeAction:	10
setInput:	10
setValue:forVariableWithName:	11
valueForVariableWithName:	11
writeToURL:error:	12

---

## **Document Revision History 13**

---

## **Index 15**

---



# AMWorkflow Class Reference

---

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

## Overview

The `AMWorkflow` class lets you use an Automator workflow in your application. You can display a workflow with an instance of `AMWorkflowView` and control its operation with an instance of `AMWorkflowController`.

A workflow consists of one or more actions (discrete tasks), which together can perform complex automation tasks. Your application can use workflows to package its own features and to take advantage of features provided by other applications. You create actions with Xcode, while you create workflows with the Automator application.

You can load and run a workflow with minimal overhead by using the `AMWorkflow` class method `runWorkflowAtURL:withInput:error:` (page 7). However, in situations where you need greater control, such as the ability to start and stop the workflow, you can use an instance of the `AMWorkflowController` class instead. In that case, you'll have to create and initialize both a workflow and a workflow controller object.

In either case, the 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 the workflow.

## Tasks

### Running a Specified Workflow

- + [runWorkflowAtURL:withInput:error:](#) (page 7)  
Loads and runs the specified workflow file.

### Creating and Initializing a Workflow

- [initWithContentsOfURL:error:](#) (page 8)  
Creates and initializes a workflow based on the contents of the specified file.

### Saving Changes to a Workflow

- [writeToURL:error:](#) (page 12)  
Writes the workflow to the specified file.

### Getting Information About the Workflow

- [actions](#) (page 7)  
Returns an array of the workflow's actions.
- [fileURL](#) (page 8)  
Returns a URL that specifies the location of the workflow file.
- [input](#) (page 9)  
Returns the input data that is passed to the first action in the workflow.
- [setInput:](#) (page 10)  
Sets the input data that is passed to the first action in the workflow.
- [setValue:forVariableWithName:](#) (page 11)  
Sets the value of the workflow variable with the specified name.
- [valueForVariableWithName:](#) (page 11)  
Returns the value of the workflow variable with the specified name.

### Manipulating the Workflow's Actions

- [addAction:](#) (page 8)  
Adds the specified action at the end of the receiving workflow.
- [insertAction:atIndex:](#) (page 9)  
Inserts the specified action at the specified position of the receiving workflow.
- [moveActionAtIndex:toIndex:](#) (page 10)  
Moves the action from the specified start position to the specified end position in the receiving workflow.

- `removeAction:` (page 10)  
Removes the specified action from the receiver.

## Class Methods

### **runWorkflowAtURL:withInput:error:**

Loads and runs the specified workflow file.

```
+ (id)runWorkflowAtURL:(NSURL *)fileURL withInput:(id)input error:(NSError **)error
```

#### **Parameters**

*fileURL*

URL that specifies the location of a workflow file.

*input*

Specifies the input for the first action in the workflow. Pass `nil` if the first action doesn't need input.

*error*

If no workflow is found or if an error occurs in initializing or running it, upon return contains an instance of `NSError` that describes the problem.

#### **Return Value**

On error, returns `nil`. Otherwise, returns the output of the last action in the workflow. Your application may need to convert the data to a desired type.

#### **Discussion**

Use this method to run a workflow without the overhead of performing a separate allocation, setting up a workflow controller, and so on. However, in situations where you need greater control, such as the ability to start and stop the workflow, use an instance of the `AMWorkflowController` class instead.

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

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`AMWorkflow.h`

## Instance Methods

### **actions**

Returns an array of the workflow's actions.

```
- (NSArray *)actions
```

**Return Value**

An array of actions for the workflow file. Actions are instances of classes such as `AMBundleAction`, `AMAppleScriptAction`, and `AMShellScriptAction`.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`AMWorkflow.h`

**addAction:**

Adds the specified action at the end of the receiving workflow.

```
- (void)addAction:(AMAction *)action
```

**Parameters**

*action*

The action to add.

**Discussion**

The workflow retains the action but does not copy it.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`AMWorkflow.h`

**fileURL**

Returns a URL that specifies the location of the workflow file.

```
- (NSURL *)fileURL
```

**Return Value**

URL that specifies the location of the workflow file.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`AMWorkflow.h`

**initWithContentsOfURL:error:**

Creates and initializes a workflow based on the contents of the specified file.

```
- (id)initWithContentsOfURL:(NSURL *)fileURL error:(NSError **)outError
```



**Parameters***fileURL*

URL that specifies the location of a workflow file.

*outError*If the workflow file can't be found, or if an error occurs in initializing the workflow, upon return contains an instance of `NSError` that describes the problem.**Return Value**

The initialized workflow object. On error, returns nil.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

AMWorkflow.h

**input**

Returns the input data that is passed to the first action in the workflow.

- (id)input

**Return Value**

The input for the first action in the workflow. Should be a data type the action can use, or a type that can be converted to one the action can use.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**- [setInput:](#) (page 10)**Declared In**

AMWorkflow.h

**insertAction:atIndex:**

Inserts the specified action at the specified position of the receiving workflow.

- (void)insertAction:(AMAction \*)*action* atIndex:(NSUInteger)*index***Parameters***action*

The action to insert.

*index*

The position in the receiver at which to insert the action. If the position is invalid, this method does nothing.

**Discussion**

The workflow retains the action but does not copy it.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

AMWorkflow.h

**moveActionAtIndex:toIndex:**

Moves the action from the specified start position to the specified end position in the receiving workflow.

```
- (void)moveActionAtIndex:(NSUInteger)startIndex toIndex:(NSUInteger)endIndex
```

**Parameters***startIndex*

The start position of the action to move.

*endIndex*

The end position for the action that is moved.

**Discussion**

If either index is invalid, this method does nothing.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

AMWorkflow.h

**removeAction:**

Removes the specified action from the receiver.

```
- (void)removeAction:(AMAction *)action
```

**Parameters***action*

The action to be removed.

**Discussion**

The action receives an `AMAction` `closed` message before being released.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

AMWorkflow.h

**setInput:**

Sets the input data that is passed to the first action in the workflow.

```
- (void)setInput:(id)input
```

**Parameters***input*

The input for the first action in the workflow. Should be a data type the action can use, or a type that can be converted to one the action can use.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [input](#) (page 9)

**Declared In**

AMWorkflow.h

**setValue:forVariableWithName:**

Sets the value of the workflow variable with the specified name.

```
- (BOOL)setValue:(id)value forVariableWithName:(NSString *)variableName
```

**Parameters***value*

The value to set for the named variable.

*variableName*

The name of a variable to set the value for.

**Return Value**

YES if *variableName* was found and its value set; otherwise NO.

**Discussion**

This method does nothing if the variable specified by *variableName* is not found.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

AMWorkflow.h

**valueForVariableWithName:**

Returns the value of the workflow variable with the specified name.

```
- (id)valueForVariableWithName:(NSString *)variableName
```

**Parameters***variableName*

The variable name.

**Return Value**

The value for the variable. Returns `nil` if no variable is found with the specified name.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setValue:forVariableWithName:](#) (page 11)

**Declared In**

AMWorkflow.h

**writeToURL:error:**

Writes the workflow to the specified file.

```
- (BOOL)writeToURL:(NSURL *)fileURL error:(NSError **)outError
```

**Parameters**

*fileURL*

URL that specifies the file location to write the workflow to.

*outError*

If the workflow file can't be written, upon return contains an instance of `NSError` that describes the problem.

**Return Value**

YES if the workflow was successfully written; otherwise NO.

**Discussion**

You might want to save the workflow, for example, because you have made changes to a variable it contains.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setValue:forVariableWithName:](#) (page 11)

**Declared In**

AMWorkflow.h

# Document Revision History

---

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

Date	Notes
2006-10-26	New document that describes the class used to run an Automator workflow in an application.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

actions [instance method 7](#)  
addAction: [instance method 8](#)

## F

---

fileURL [instance method 8](#)

## I

---

initWithContentsOfURL:error: [instance method 8](#)  
input [instance method 9](#)  
insertAction:atIndex: [instance method 9](#)

## M

---

moveActionAtIndex:toIndex: [instance method 10](#)

## R

---

removeAction: [instance method 10](#)  
runWorkflowAtURL:withInput:error: [class method 7](#)

## S

---

setInput: [instance method 10](#)  
setValue:forVariableWithName: [instance method 11](#)

## V

---

valueForVariableWithName: [instance method 11](#)

## W

---

writeToURL:error: [instance method 12](#)