
Amazon WorkSpaces Application Manager Administrator's Guide



Amazon WorkSpaces Application Manager: Administrator's Guide

Copyright © 2015 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, AWS CloudTrail, AWS CodeDeploy, Amazon Cognito, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Amazon Kinesis, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC, and Amazon WorkDocs. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is Amazon WAM	1
Related Services	1
Setting Up	2
AWS Account	2
Amazon WorkSpaces	2
Controlling Access	3
Application Packaging Role	3
AWS Marketplace Access Role	3
Amazon WAM Access Role	4
Getting Started	5
Sign Up for Amazon WAM	5
Build Your Application Catalog	5
Assign Applications to Your Users	6
Amazon WorkSpaces Application Manager Client Application	6
Managing Applications	7
Building Your Application Catalog	7
Subscribing to an Application	7
Uploading Your Own Applications	8
Removing an Application	9
Assigning an Application	11
Updating an Application	12
Versions	13
Constraints	14
Tags	14
Users and Groups	14
Support Details	15
Monitoring Application Usage	15
Applications	15
Users	15
Removing All Application Assignments	16
Troubleshooting	16
I assigned an application to a user, but I don't see the user in the usage report	16
Packaging and Validating	17
Packaging an Application	17
Best Practices	17
Launching the Packaging Instance	18
Creating a New Package	18
Creating a New Application Version	24
Packaging Specific Applications	25
Validating a Package	30
Launching the Validation Instance	30
Validating the Application Package	30
Troubleshooting	31
Application Packaging Issues	31
Device Driver and Service Issues	34
I am missing shortcuts, files, and folders.	37
Other Issues	38
User Account Issues	39
Limits	40
Document History	41

What is Amazon WAM

Amazon WorkSpaces Application Manager (Amazon WAM) offers a fast, flexible, and secure way for you to deploy and manage applications for [Amazon WorkSpaces](#). Amazon WAM accelerates software deployment, updates, patching, and retirement by packaging Microsoft Windows desktop applications into virtual containers that run as though they are installed natively.

Amazon WAM is fully integrated with the AWS Management Console, and allows you to build an application catalog from your line-of-business applications, third-party applications that you own the license for, and applications purchased through the AWS Marketplace. Simple, pay-as-you-go, user-based pricing ensures that you only pay for the applications that are used.

Amazon WAM is available in two versions: Amazon WAM Lite and Amazon WAM Standard. With Amazon WAM Lite, you can manage and deliver applications from the AWS Marketplace free of charge. You pay only for the applications that your users activate. With Amazon WAM Standard, you can build your application catalog with line-of-business applications, third-party applications for which you own licenses, and applications from the AWS Marketplace for Desktop Apps. You also gain access to additional features.

Related Services

Amazon WAM is only available on [Amazon WorkSpaces](#). For more information, see [Amazon WorkSpaces FAQ](#).

Setting Up Amazon WAM

To use Amazon WorkSpaces Application Manager (Amazon WAM), you must meet the following prerequisites.

Topics

- [AWS Account](#) (p. 2)
- [Amazon WorkSpaces](#) (p. 2)
- [Controlling Access to Amazon WAM Resources](#) (p. 3)

AWS Account

Your AWS account gives you access to all services, but you are charged only for the resources that you use.

If you do not have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open <http://aws.amazon.com/> and click **Sign Up**.
2. Follow the on-screen instructions.

Your root account credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your WorkSpaces. To allow other users to manage AWS Directory Service resources without sharing your security credentials, use AWS Identity and Access Management (IAM). We recommend that everyone work as an IAM user, even the account owner. You should create an IAM user for yourself, give that IAM user administrative privileges, and use it for all your work.

Amazon WorkSpaces

Amazon WAM requires that you have an Amazon WorkSpaces directory and users. Your users access the Amazon WAM applications that are assigned to them on Amazon WorkSpaces, so any user that you assign an application to must also have a WorkSpace provisioned for them. For more information, see the [Amazon WorkSpaces Administration Guide](#).

Controlling Access to Amazon WAM Resources

By default, IAM users don't have permission to Amazon WAM resources. Amazon WAM must have permission to perform certain actions on your behalf. To perform these actions, you must create three IAM roles that grant Amazon WAM permission to perform actions on your behalf.

Topics

- [Application Packaging Role \(p. 3\)](#)
- [AWS Marketplace Access Role \(p. 3\)](#)
- [Amazon WAM Access Role \(p. 4\)](#)

Application Packaging Role

This IAM role allows the Amazon WAM packaging instance to access your application package catalog. If you have not already done so, create the `AmazonWamAppPackaging` role by performing the following steps.

To create an IAM role to access your Amazon WAM application catalog

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Roles**, and then choose **Create New Role** to start the wizard.
3. Enter `AmazonWamAppPackaging` in the **Role Name** field and choose **Next Step**.

Important

The name of the role must be `AmazonWamAppPackaging`. If it is not, the packaging and validation applications will not be able to access your packages.

4. Under **Select Role Type**, choose **AWS Service Roles**, and then choose **Select** next to **Amazon EC2**.
5. Under **Attach Policy**, select **AmazonWorkSpacesApplicationManagerAdminAccess** and choose **Next Step**.
6. Review the policy statement to confirm that you are granting permission to Amazon WAM to perform the actions listed. After reviewing and accepting the policy and permissions, click **Create Role**.

AWS Marketplace Access Role

This IAM role allows Amazon WAM to access the AWS Marketplace on your behalf. The first time you log in to the Amazon WAM console, you will be asked if you want to create a role with the name `AmazonWamMarketplace_Default_Role`. In order to use Amazon WAM, you must allow this role to be created.

The following is the IAM policy for the `AmazonWamMarketplace_Default_Role` role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "catalog-admin:Describe*",
        "catalog-admin:Get*",
        "catalog-admin:Search*",
        "catalog-admin:List*"
      ]
    }
  ]
}
```

```
        "catalog-admin:CreateListing",
        "catalog-admin:UpdateListing",
        "catalog-admin>DeleteListing",
        "catalog-user:SimulateView*",
        "catalog-user:SimulateGet*",
        "catalog-user:SimulateBrowse*"
    ],
    "Resource": [
        "*"
    ]
}
]
```

Amazon WAM Access Role

This IAM role allows a group or user to administer Amazon WAM.

The following is the IAM policy for this role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "catalog-admin:*",

        "ds:*",
        "iam:ListAttachedRolePolicies",
        "iam:ListRoles",
        "iam:CreateRole",
        "iam:PutRolePolicy"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Getting Started with Amazon WAM

The following topics explain how to get started using Amazon WorkSpaces Application Manager (Amazon WAM).

Topics

- [Sign Up for Amazon WAM \(p. 5\)](#)
- [Build Your Application Catalog \(p. 5\)](#)
- [Assign Applications to Your Users \(p. 6\)](#)
- [Amazon WorkSpaces Application Manager Client Application \(p. 6\)](#)

Sign Up for Amazon WAM

Amazon WAM is available to all Amazon WorkSpaces customers. After you finish setting up Amazon WorkSpaces, you can start using Amazon WAM console by going to the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and then choosing **Application catalog** in the navigation bar. The first time you open the Amazon WAM console, two Amazon EC2 AMIs are shared with your account:

Amazon WAM Studio AMI

This AMI contains the Amazon WorkSpaces Application Manager Studio application to package your own applications.

Amazon WAM Admin Player AMI

This AMI contains the Amazon WAM Admin Player application to validate your application packages.

Amazon WAM will publish new AMIs when they become available. You will automatically receive the latest AMIs to replace your older version.

Build Your Application Catalog

You can build your application catalog from three sources:

- The AWS Marketplace
- Your line-of-business applications
- Applications that you already own the license for

For more information about how to get started building your application catalog, see [Building Your Application Catalog \(p. 7\)](#).

Assign Applications to Your Users

Your users only have access to the Amazon WAM applications that you assign to them. For more information about assigning applications to your users, see [Assigning an Application to Users and Groups \(p. 11\)](#).

Amazon WorkSpaces Application Manager Client Application

Your users use the Amazon WorkSpaces Application Manager client application on their WorkSpace to access the applications that you assign to them. The Amazon WorkSpaces Application Manager client application installer is available on your user's WorkSpace desktop. If the installer is missing from your user's WorkSpace, it will be added the next time the user reboots the WorkSpace. Your users must install the Amazon WorkSpaces Application Manager client application to gain access to the applications that you assign to them. For more information about using the Amazon WorkSpaces Application Manager client application, see the [Amazon WAM User Guide](#).

Managing Your Amazon WAM Applications

The following topics describe how to manage your Amazon WorkSpaces Application Manager (Amazon WAM) applications.

Topics

- [Building Your Application Catalog \(p. 7\)](#)
- [Assigning an Application to Users and Groups \(p. 11\)](#)
- [Updating an Application \(p. 12\)](#)
- [Monitoring Application Usage \(p. 15\)](#)
- [Removing All Application Assignments \(p. 16\)](#)
- [Troubleshooting Amazon WAM Application Issues \(p. 16\)](#)

Building Your Application Catalog

All of your Amazon WAM applications are listed in your application catalog, which contains all of the applications that you can assign to your users. This includes both applications that you subscribe to from the AWS Marketplace, as well as applications that you upload.

Topics

- [Subscribing to an Application From AWS Marketplace \(p. 7\)](#)
- [Uploading Your Own Applications \(p. 8\)](#)
- [Removing an Application From Your Application Catalog \(p. 9\)](#)

Subscribing to an Application From AWS Marketplace

In addition to your own application, you can subscribe to applications from the AWS Marketplace with a monthly subscription.

To subscribe to an application from the AWS Marketplace

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Application catalog** in the navigation pane.
2. Choose **Add application from AWS Marketplace**.
3. Select the application to add to your catalog.
4. In the application detail page, select the version or versions of the application to add to your catalog and choose **Accept terms and subscribe**.
5. In the application list page, choose **Return to my application catalog**.
6. If needed, set **Source** to **AWS Marketplace**. The application is added to the list of subscribed applications.

Uploading Your Own Applications

To make your own applications available in your application catalog, you create a new application from an application package. For more information about creating an application package, see [Packaging an Application with Amazon WAM \(p. 17\)](#).

To create a new application from an application package

Before beginning this procedure, you must have at least one validated application package.

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Application catalog** in the navigation pane.
2. Choose **Upload new application** and perform the following steps:
3. In the **Enter application details** page, enter the following fields and choose **Next**.

Application name

Enter the name of the application.

Short description

Enter a short description of the application.

Description

Enter a description of the application.

Category

Select the appropriate application category.

Provided by

Enter the name to display as the provider of the application.

Vendor

Enter the name to display as the vendor of the application.

4. In the **Enter support details** page, enter the following fields and choose **Next**. All of these fields are optional.

Email contact

Enter the email address users should use to obtain support for the application.

Support link

Enter the URL that users should use to obtain support for the application.

Support description

Enter any additional support information.

5. In the **Select application package** page, enter the following fields and choose **Review**.

Package details

Package title

Select the title of the package to create the application from. The list contains all of your validated application packages.

Package build

Select the package build for the application.

Version information

Version title

Enter a short name for the version.

Description

Enter the description of the version.

Application licensing

License type

Select the license type for the application.

By WorkSpace

The license is per WorkSpace.

By user

The license is per user.

Application constraints

Maximum install

Enter the maximum number of installations allowed for the application.

6. In the **Review** page, review the application information, make any necessary changes, and choose **Confirm and upload**. The application may not be immediately available.

Removing an Application From Your Application Catalog

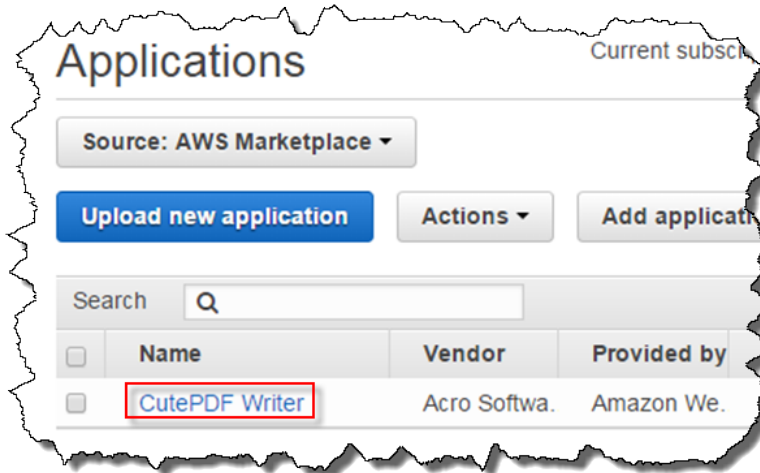
You may want to remove an application from Amazon WorkSpaces because you want to deprecate an existing application for a different application or your user has stopped using an existing application. You can remove any application from AWS Marketplace application or an uploaded application from your application catalog.

Important

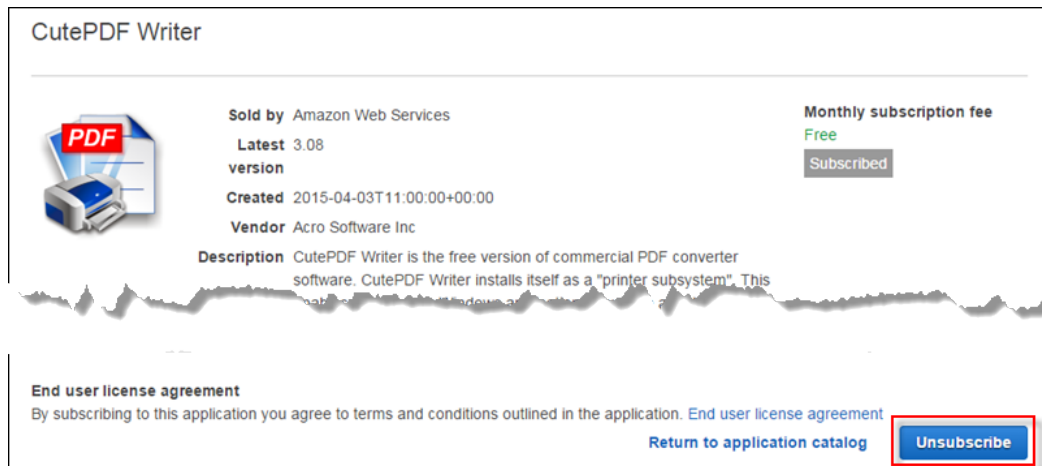
Before you can remove an application, you must remove all assignments for the application. For more information, see [Users and Groups \(p. 14\)](#).

To remove an AWS Marketplace application from your application catalog

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Application catalog** in the navigation pane.
2. Set **Source** to **AWS Marketplace** and choose the application to remove from your application catalog. The link opens the application product page.

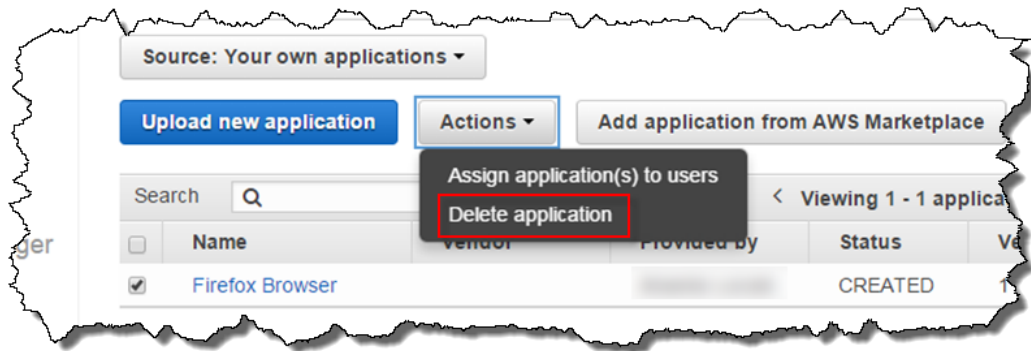


3. In the application product page, choose **Unsubscribe**.



To remove an application you uploaded from your application catalog

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Application catalog** in the navigation pane.
2. Set **Source** to **Your own applications**.
3. Choose the check box for the application you want to remove from your application catalog.
4. Choose **Actions** and select **Delete application**.



Assigning an Application to Users and Groups

To assign an application to users or groups, perform the following steps:

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Application catalog** in the navigation pane.
2. If the application is an uploaded application, set **Source** to **Your own applications**. If the application is a subscribed application, change **Source** to **AWS Marketplace**.

Note

You will only be able to assign your uploaded applications on Amazon WAM Standard.

3. Select up to five applications to assign, choose **Actions**, and select **Assign application(s) to users**.
4. In the **Select users** page, make the following selections and choose **Search**.

Directory

Select the AWS Directory Service directory in which to search for users or groups.

Type

Select the type of entity for which to search, users or groups.

Fields containing

Enter all or part of the user or group name for which to search. Leave this field blank to search for all users or groups.

If there are more search results than can be displayed in one page, you can scroll through the pages by clicking on the left arrow (<) or right arrow (>) in the list header.

5. From the **Search results** list, select the user or group to which to assign the application, and choose the right arrow (>) between the two lists. The selected items are moved to the **Selected** list.

To remove an item from the **Selected** list, select the item and choose the left arrow (<) between the two lists. The selected items are moved to the **Search results** list.

You can change the search criteria at any time to add other users or groups to the **Selected** list. The items in the **Selected** list are not removed when you redo the search.

Repeat this step until all of the users and groups to which to assign the application are selected. When you are ready, choose **Next**.

6. In the **Configure assignment options** page, make the following selections, and choose **Review**.

Note

These selections are only available in Amazon WAM Standard. Amazon WAM Lite has limited selections.

Version

Select the application version to assign.

Installation Type

Select one of the following:

Optional

The application will be installed when the user initiates the installation from the Amazon WorkSpaces Application Manager client application.

Required

The application will automatically be installed on the user's WorkSpace.

Auto update

Select one of the following:

No

You need to assign application updates manually to your users.

Yes

The application is updated automatically when a new version of the application is published.

7. In the **Review** page, review the assignments, make any changes needed, and choose **Confirm and assign**. The selected applications are assigned to the selected users and group.

Updating an Application

From the application detail page, you can perform the following tasks:

- View and edit the application versions, and create a new version
- View and edit the constraints for the application
- View and edit the tags for the application
- View and edit the assignments for the application
- View the support details for the application

Amazon WAM Lite customers will have limited features. Amazon WAM Standard customers will be able to assign users to the applications you uploaded and configure the assignment options such as controlling versions, managing the update settings, or configuring the installation type.

To open the detail page for an application

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Application catalog** in the navigation pane.
2. To view the details for one of your own applications, set **Source** to **Your own applications**. To view the details for a subscribed application, change **Source** to **AWS Marketplace**.
3. Open the application detail page by choosing the name of the application.
4. If the application is one of your own applications, you can edit the information about the application, such as the name, vendor, provider, and description. To edit these, choose the pencil icon next to the application name, enter the new information, and choose **Save**.

Topics

- [Versions \(p. 13\)](#)
- [Constraints \(p. 14\)](#)
- [Tags \(p. 14\)](#)
- [Users and Groups \(p. 14\)](#)

- [Support Details \(p. 15\)](#)

Versions

You can view the version information for all of the applications in your catalog, regardless of the source. You can also update the versions for your own applications. To view the versions for an application, in the application detail page, expand the **Versions** section.

If the application is one of your own applications, you can perform the following tasks:

- [Create a New Version \(p. 13\)](#)
- [Edit an Existing Version \(p. 13\)](#)

Amazon WAM Lite subscribers will not be able assign any versions of the uploaded applications non any versions of the applications from the AWS Marketplace for Desktop Apps that are not the latest version.

Create a New Version

You can create a new version of an application. Before you can create a new version of the application, you must have uploaded a new version of the application package as explained in [Creating a New Version of an Application \(p. 24\)](#).

To create a new version of an application

1. In the **Versions** section of the application details page, choose **Create new version**.
2. In the **Select application package** page, enter the following fields and choose **Submit**. The new application version is created.

Package details

Package build

Select the package build for the new version.

Version information

Version title

Enter a short name for the new version.

Description

Enter the description of the new version.

Edit an Existing Version

You can modify the title and description of an existing version.

To edit an existing version of an application

1. In the **Versions** section of the application details page, select the version to edit, and choose **Edit version**.
2. Modify the version title and description as required and choose **Submit**. It takes several minutes for the updated version information to be available.

Constraints

You can view and modify the constraints for an application from the application details page.

To view the constraints for an application

1. In the application detail page, expand the **Constraints** section. The application constraints are displayed.
2. Here, you can perform the following tasks:
 - To add a constraint, choose **Add constraints**.
 - To create a new version of the application, choose **Create new version**, and perform the procedure in [Assigning an Application to Users and Groups \(p. 11\)](#).
 - To edit an existing version of the application, select the version to edit and choose **Edit version**. You can modify the version title and description.

Tags

Amazon WAM Standard customers can view, add, and remove tags from your applications.

To view the tags for an application

1. In the application detail page, expand the **Tags** section. The application tags are displayed.
2. Here, you can perform the following tasks:
 - To add a tag, enter the tag name and value, and choose **Add tag**.
 - To remove a tag, select the tag and choose **Delete tag**.

Users and Groups

You can view and modify the users and groups to which your applications are assigned. Amazon WAM Lite customers will be unable to assign users and groups to your uploaded applications. Amazon WAM Standard customers will be able to assign users and groups to your uploaded applications.

To view the users and groups to which an application is assigned

1. In the application detail page, expand the **Users and Groups** section. The users to which the application is assigned are displayed.
2. Here, you can perform the following tasks:
 - To assign the application to another user or group, choose **Add user or group**, and perform the procedure in [Assigning an Application to Users and Groups \(p. 11\)](#).
 - To update the assignment options for the application, choose **Edit assignment options**. On the [Configure assignment options \(p. 11\)](#) page, you can update the assignment options.
 - To remove an application assignment, select the users and groups to remove the assignment from, choose **Remove user or group**, verify that you want to perform the action, and choose **Continue**. The application is no longer assigned to the selected users and groups.

Support Details

To view the support information for an application, in the application detail page, expand the **Support details** section. The support information for the application is displayed.

Monitoring Application Usage

Amazon WAM allows you to track your application usage by viewing usage reports in the Amazon WAM console.

Topics

- [Applications \(p. 15\)](#)
- [Users \(p. 15\)](#)

Applications

To view application usage information, perform the following steps.

To view application usage information

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Usage** in the navigation pane.
2. In the **Usage** page, choose the **Application** tab. All of your applications are displayed in the list.

If there are more search results than can be displayed in one page, you can scroll through the pages by choosing the left arrow (<) or right arrow (>) in the list header.

You can view additional information about a specific application by choosing the expander button next to the product.

Users

To view usage information by user, perform the following steps.

To view usage information by user

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Usage** in the navigation pane.
2. In the **Usage** page, choose the **Users** tab.
3. Under **Directory**, choose your directory. All of your users that have assigned applications are displayed in the list.

Note

Users are not displayed in the list until they have connected to their Workspace after the Amazon WorkSpaces Application Manager client application has been installed.

If there are more search results than can be displayed in one page, you can scroll through the pages by choosing the left arrow (<) or right arrow (>) in the list header.

You can view the applications that are assigned to a specific user by choosing the expander button next to the username.

Removing All Application Assignments

The Amazon WAM service automatically registers itself with an AWS Directory Service directory. This allows Amazon WAM to access the directory and prevents the directory from being deleted while it is in use by Amazon WAM. Before you can delete the directory that your WorkSpaces are assigned to, you must remove all Amazon WAM application assignments in the directory and deregister Amazon WAM from the directory. To make this task easier, the Amazon WAM console provides the ability to remove all application assignments and deregister itself from a directory in a single operation.

Note

Even if you do not have any application assignments, you must perform the following procedure to deregister Amazon WAM from a directory.

To remove all application assignments and deregister Amazon WAM from a directory

1. Open the Amazon WorkSpaces console at <https://console.aws.amazon.com/workspaces/> and select **Usage** in the navigation pane.
2. In the **Usage** page, choose the **Users** tab.
3. Under **Directory**, choose the directory. All of the directory users that have assigned applications are displayed in the list.

Note

Users are not displayed in the list until they have connected to their WorkSpace after the Amazon WorkSpaces Application Manager client application has been installed.

4. Choose **Remove all assignments**. The process to remove all application assignments and deregister the directory is initiated immediately. The entire process takes 10 to 15 minutes to complete. After all applications assignments have been removed, the service is deregistered from the directory, which is removed from the list of directories in the **Users** tab.

Troubleshooting Amazon WAM Application Issues

Topics

- [I assigned an application to a user, but I don't see the user in the usage report \(p. 16\)](#)

I assigned an application to a user, but I don't see the user in the usage report

You won't see the user in the usage report until the user has launched the Amazon WorkSpaces Application Manager client application for the first time. For more information, see [Users \(p. 15\)](#).

Packaging and Validating Your Applications

To create Amazon WorkSpaces Application Manager (Amazon WAM) applications of your own making, you must create the application package, and validate that the package installs and works correctly. This is accomplished using two special EC2 instances.

Topics

- [Packaging an Application with Amazon WAM \(p. 17\)](#)
- [Validating an Application Package with Amazon WAM \(p. 30\)](#)
- [Troubleshooting Amazon WAM Packaging Issues \(p. 31\)](#)

Packaging an Application with Amazon WAM

An application package is created on a special EC2 instance. The first time you access the Amazon WAM console, an Amazon EC2 AMI is shared with you. Use this AMI to launch the packaging instance.

Topics

- [Best Practices for Application Packaging \(p. 17\)](#)
- [Launching the Packaging Instance \(p. 18\)](#)
- [Creating a New Application Package \(p. 18\)](#)
- [Creating a New Version of an Application \(p. 24\)](#)
- [Packaging Specific Applications \(p. 25\)](#)

Best Practices for Application Packaging

The following are the recommended best practices to use when packaging an application:

- Always launch a new packaging instance when creating a new application package.
- For best performance, we recommend using an m3.xlarge or better instance type for the packaging instance. Normal Amazon EC2 instance charges apply.
- Always review the virtualization settings after capturing the application installation. For more information, see [Installation Capture Information \(p. 21\)](#).

Launching the Packaging Instance

We recommend that you launch the packaging and validation instances into their own VPC to make launching and configuration easier. For this, you need the following:

- A VPC in a region that Amazon WAM is supported in.
- A public subnet in the VPC.

Before launching the packaging instance, you must create the `AmazonWamAppPackaging` IAM role. For more information, see [Application Packaging Role \(p. 3\)](#).

To launch a packaging instance, follow the steps in the [Launching Your Instance from an AMI](#) topic in the *Amazon EC2 User Guide for Microsoft Windows Instances*. The following are specific settings for launching a packaging instance:

- In step 2, select the region in which your VPC is located.
- In step 4, select the **My AMIs** tab, select **Shared with me**, and select the **Amazon WAM Studio AMI**.
- In step 5, for best performance, we recommend **m3.xlarge** or better for the instance type. You will be charged for AWS resources such as the EC2 instance, Amazon S3 storage, and bandwidth when using this instance to package your applications.
- In step 6, on the **Configure Instance Details** page, enter the following:
 - Under **Network**, select your VPC.
 - Under **Subnet**, select the public subnet in the VPC.
 - Under **IAM role**, select the **AmazonWamAppPackaging** role, which you created in [Application Packaging Role \(p. 3\)](#).
- In step 9, on the **Configure Security Group** page, select a security group that allows access to the instance over port 3389 (RDP) from your network.

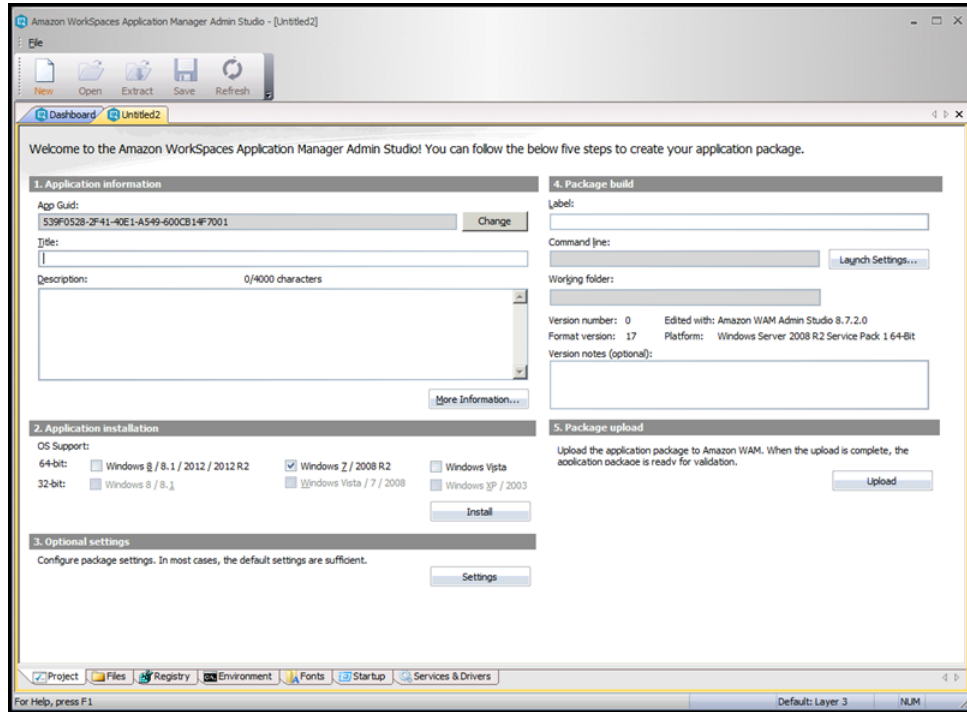
Creating a New Application Package

After the packaging instance is launched, use it to create an application package. AWS recommends that you launch an entirely new packaging instance for each application package that is created.

To package an application

1. Using a Remote Desktop client, connect to the packaging instance. The remaining instructions are performed on the packaging instance.
2. Download the installer for the application to be packaged to the instance. All Amazon WAM applications run on Amazon WorkSpaces. If the application to be packaged is available for different platforms, you should download the installer for Windows.
3. Launch the Amazon WorkSpaces Application Manager Studio by opening the desktop shortcut.

Amazon WorkSpaces Application Manager Administrator's Guide Creating a New Package



4. In the Amazon WorkSpaces Application Manager Studio toolbar, click **New** to create a new application project. Fill in information as noted below.

To package an application, enter the following information:

- [Application settings](#) (p. 19)
- [Application installation](#) (p. 19)
- [Optional settings](#) (p. 22)
- [Package build](#) (p. 23)
- [Package upload](#) (p. 23)

Application settings

Enter information in the following fields:

App Guid

The identifier of the application. This is used by Amazon WAM to identify a unique application package for troubleshooting and debugging. An identifier is automatically created when you create a new application project. Choose **Change** to generate a new identifier if needed.

Title

The name of the application.

Description

A description of the application.

Application installation

Select the target operating system for the application and capture the application installation by performing the steps in [Capturing the Application Installation](#) (p. 20).

Capturing the Application Installation

When creating an application package, you install the application on the packaging instance. During the installation, the Amazon WorkSpaces Application Manager Studio application captures all changes to the file system, registry, environment, and other systems during the installation.

Before capturing the application installation, save the application project. This is required before you can capture the application installation because some applications require that the system be restarted to complete.

The installation process is different for [applications that have an installer \(p. 20\)](#) and [applications that do not have an installer \(p. 21\)](#).

Topics

- [Applications With an Installer \(p. 20\)](#)
- [Applications Without an Installer \(p. 21\)](#)
- [Captured Information \(p. 21\)](#)

Applications With an Installer

This procedure is used for applications that have an installer.

To capture the installation of an application that has an installer

1. Click **Install**.
2. Enter the following fields in the **Capture Application Installation** dialog box:

Installer path

Enter the path of the application installer.

Command line parameters

Enter any command line parameters needed for the installation.

Compatibility Settings

Click **Compatibility Settings** and enter any Windows compatibility settings that are required by the installer, such as being run as administrator.

Capture the file and registry activity for the following processes

Installer process and sub-processes only (default)

Only changes made by the installer process, and any processes the installer creates, are captured.

All processes

Changes made by all processes are captured.

Ignore changes under the Installer path specified above

File changes made to the folder that contains the installer are not captured. This is useful if the installation process creates temporary files in the installer directory that are not required by the installed application.

After capturing is complete, do the following actions during the merge:

Replace short path names with long ones in the registry

Any short (8.3 format) path names captured in the registry are replaced with long path names, if possible.

3. Click **Launch** and install the application as needed. All changes to the files system, registry, and environment are captured during the installation. For more information about what is captured, and how to modify that information, see [Captured Information \(p. 21\)](#).
4. When the installation is complete, click **OK** to close the **Capture** dialog box.

Applications Without an Installer

This procedure is used for applications that do not have an installer, such as applications that are delivered as standalone executables, or are delivered in an archive, such as a ZIP file.

To capture the installation of an application that does not have an installer

1. Click **Install**.
2. Enter the following fields in the **Capture Application Installation** dialog box:

Installer path

Enter `explorer.exe`.

Command line parameters

Leave this field empty.

Compatibility Settings

Do not modify these settings.

Capture the file and registry activity for the following processes

Select **All processes**.

Ignore changes under the Installer path specified above

Leave this set to the default setting.

After capturing is complete, do the following actions during the merge:

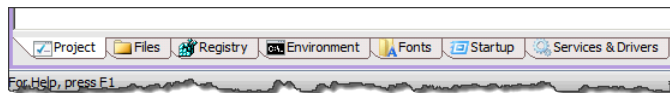
Replace short path names with long ones in the registry

Leave this set to the default setting.

3. Click **Launch**. A Windows Explorer window is opened.
4. Perform all steps needed to install the application, such as copying the standalone executable to the desired location, or extracting the files from a ZIP archive. All changes to the files system are captured during the installation. For more information about what is captured, and how to modify that information, see [Captured Information](#) (p. 21).
5. Close the Windows Explorer window and click **OK** to close the **Capture** dialog box.

Captured Information

The tabs at the bottom of the Amazon WorkSpaces Application Manager Studio contain all of the information that was captured during the installation of the application. Review each of these and modify them as needed for your package. Care must be taken when modifying these as making changes can make the application inoperable.



Installation Capture Information

Files

All changes to the file system are contained here. You can add, remove, or move files to customize your package.

Registry

All changes to the system registry are contained here. You can add, remove, or move registry entries to customize your package.

Environment

Any changes to the environment variables are contained here. You can add, remove, or modify these variables to customize your package.

Fonts

All changes to the system fonts are contained here. You can add or remove fonts to customize your package.

Startup

All changes to the system startup folders and registry keys are contained here. You can add, remove, or modify these to customize your package.

Services & Drivers

All services and drivers that were installed during the installation are contained here. You can add or remove these to customize your package.

Optional settings

Choose **Settings** and enter any packager settings that are required. In most cases, the default settings are used.

Paging

Specify any additional sources for the application package.

Configurable AppEvents

Add any additional app events needed for the package. These include any actions you want to take place after certain application events, such as application launch or registration.

Virtualization

Add settings and options that affect how the application is virtualized. An application is virtualized when it is installed on the client.

Environment Variables

Specify if the package environment variables are available to the entire system, or just to the application.

Fonts

Specify if the package fonts are available to the entire system, or just to the application.

Startup Items

Specify if the startup folder and registry keys are automatically run when the application is virtualized.

Special Requirements

Specify additional actions needed when the application is virtualized.

Sandboxing

Choose how the application is sandboxed on the client.

Folder Exclusions

Add any folders to exclude from the package.

Registry Key Exclusions

Add any registry keys to exclude from the package.

Disposition for new registry key roots

Specifies what access the local system has to the application's resources, such as files, registry settings, and so on.

Virtual-integrated (layer 3)

Resources can be seen by both the application and the local system, but are not physically installed on the local system. This is the default setting.

Virtual-isolated (layer 4)

Resources can only be seen by the application, and are not physically installed on the local system.

Application settings

Enter additional settings for the application.

Preserve application settings when application is removed

Specifies if the application-specific settings modified by the user should be preserved or removed when the application assignment is removed from the user. If this field is set to

preserve these settings, the application settings will be available if the application is assigned to the user again.

Allow Windows to roam application settings

Specifies if the application-specific settings should be written to the standard Windows roaming profiles location.

Security

Add or modify how processes on the client can access the application's virtual resources.

Performance

This is the percentage of cache space the application needs to run smoothly. In most cases, the default of 10% is sufficient. This space will be allocated to the minimum size within the Amazon WorkSpaces Application Manager client application cache. Setting this field may cause the Amazon WorkSpaces Application Manager client application to prompt to increase a user's cache minimum size. After testing and optimization is completed, this additional space can be adjusted if needed.

Package build

Enter the following fields.

Label

Enter a label for the package. This is displayed in the Amazon WAM console as the version label.

Command line

Choose **Launch Settings** and enter the command to launch the installed application, along with any other launch and compatibility settings that are needed for the application.

Important

This is the command that is used to launch the application on the client, not the application installer.

To set the command used to launch the application on the client

1. Select the **Files** tab.
2. Find the application's main executable file in the list of files.
3. Right click on the file and choose **Set As Command Line**. The application command line and working folder are automatically populated to point to the selected file.

Working folder

Set this to the working folder for the application.

All packages must have a command line. If the package you are creating does not have a primary application, you should set this to `explorer.exe`.

After you upload a package, you cannot change the command line for the package. If you need to modify the command line, you need to create an entirely new package.

Package upload

Upload the package for validation by performing the following steps:

To upload the package for validation

1. Save the application project.
2. Choose **Upload** to upload the application package to the Amazon WAM service. When the upload is complete, choose **Close** in the **Progress** dialog box.

3. Close the application project. It is important that you do not modify a project after it has been uploaded. If you need to update a package, create a new version of the package. For more information, see [Creating a New Version of an Application \(p. 24\)](#).

The application package is now ready for validation. For more information, see [Validating an Application Package with Amazon WAM \(p. 30\)](#).

Creating a New Version of an Application

In addition to creating new application packages, you can also create new versions of existing packages. This is particularly useful for distributing patches and updates for your applications. AWS recommends that you launch an entirely new packaging instance for each application package version that is created. To launch a packaging instance, follow the steps in [Launching the Packaging Instance \(p. 18\)](#).

After the packaging instance is launched, use it to create a new application package version. If an application patch is being created, you can reuse a packaging instance that only has the application being patched installed on it.

Creating an Application Update

An application update (or patch) is an update to an existing application that requires that the application already be installed. AWS recommends using this approach to apply security updates, application updates, and minor version upgrades.

To create an application update

1. Using a Remote Desktop client, connect to the packaging instance. The remaining instructions are performed on the packaging instance.
2. Download the update installer or new application version to the instance.
3. Launch the Amazon WorkSpaces Application Manager Studio by opening the desktop shortcut.
4. In the Amazon WorkSpaces Application Manager Studio **Dashboard**, select the application to create an update for.
5. If the current version of the application is not already installed on the instance, choose **Extract** on the Amazon WorkSpaces Application Manager Studio toolbar. The current version of the application is installed on the instance and a new version of the application project is opened.

Otherwise, choose **Open** and a new version of the application project is opened.

6. Capture the upgrade installation as described in [Capturing the Application Installation \(p. 20\)](#).
7. If you need to modify any of the optional settings, choose **Settings** and modify any packager settings that are required. In most cases, the default settings are used. For more information, see [Optional settings \(p. 22\)](#).
8. In the **Label** field of the **Package build** section, enter the label for the new version.

Note

You cannot change the command line for a package that has been uploaded. If you need to modify the command line, you need to create an entirely new package.

9. Upload the new package version as described in [Package upload \(p. 23\)](#).

The new version of the application package is now ready for validation. The new version cannot be assigned to users until the validation is completed. For more information, see [Validating an Application Package with Amazon WAM \(p. 30\)](#).

Packaging Specific Applications

This section shows you how to package specific applications.

Topics

- [Packaging 7-Zip with Amazon WAM \(p. 25\)](#)
- [Packaging Google Chrome with Amazon WAM \(p. 26\)](#)
- [Packaging Mozilla Firefox with Amazon WAM \(p. 28\)](#)

Packaging 7-Zip with Amazon WAM

To create a package, launch a packaging instance and a validating instance on Amazon EC2 and then create and validate a new package. For more information, see the following topics:

- [Packaging an Application with Amazon WAM \(p. 17\)](#)
- [Packaging and Validating Your Applications \(p. 17\)](#)

To create a package

To create a new package, use Windows Remote Desktop to connect to your package instance and then use Amazon WorkSpaces Application Manager Studio to create the package. For more information, see [Connecting to Your Windows Instance Using RDP](#).

1. Go to the [7-Zip download page](#) and download the Windows installer for 7-Zip.
2. Double-click the desktop shortcut to start the Amazon WorkSpaces Application Manager Studio.
3. On the Amazon WorkSpaces Application Manager Studio dashboard, choose **New**.
4. Under **1. Application settings**, do the following:
 - For **Title**, enter the name of the application. This text appears in the **Dashboard** tab and in the Amazon WorkSpaces Application Manager Studio, to identify this application.
 - For **Description**, enter a short description for the application.
5. In **2. Application Installation**, select an operating system and choose **Install**.
6. In the **Capture Application Installation** dialog box, do the following:
 - For **Installer path**, enter the path to the 7-Zip installer.
 - For **Capture file and registry activity for the following processes**, select **Installer process and sub-processes only**.
 - Select **Ignore changes under the Installer path specified above**.
 - For **After capturing is complete, do the following actions during the merge**, select **Replace short path names with long ones in the registry**.
 - Choose **Launch**.

The installer for 7-Zip starts after you choose **Launch**. Follow the steps in the installer wizard to complete the installation.

7. Choose the **Files** tab and do the following:
 - In the **<ProgramFiles64>7-Zip** directory, open the context (right-click) menu for **7-zip.dll** and choose **Properties**.
 - In the **File Properties** dialog box, select **Installed temporary (layer 2)** and then choose **OK**.

8. To ensure that Windows Explorer does not hold a reference to `7-zip.dll` when the application is removed from the WorkSpace, create a configurable AppEvent that deregisters the file by using a batch file.

To create the batch file, do the following:

- On the packaging instance desktop, start Notepad.
- Type the following and save the file to the desktop as `unreg_7zip.bat`.

```
@echo off
regsvr32 /s /u "c:\Program Files\7-Zip\7-zip.dll"
```

To create the configurable AppEvent, do the following:

- In Amazon WorkSpaces Application Manager Studio dashboard, under **3. Optional settings**, choose **Settings**.
 - In the **Settings** dialog box, choose **Configurable AppEvents**.
 - In the **Configurable AppEvent Setting** dialog box, do the following:
 - Under **Trigger**, choose **Before deactivation**.
 - In **File name** under **Handler command line**, choose the `unreg_7zip.bat` file you created earlier.
 - Under **Handler launching**, select **Run elevated (only supported on Windows Vista or later)**.
 - Under **After launching**, select **Wait for the handler to exit**. In time out after, type 5 and select **minute(s)**.
 - Choose **OK**.
 - Choose **OK**.
9. Skip **3. Optional settings**.
 10. Choose **Save** in the toolbar to save the application package project.
 11. Select the **Project** tab. In **4. Package Build**, do the following:
 12. In **5. Package upload**, choose **Upload** and then **Close**.

Packaging Google Chrome with Amazon WAM

To create a package, launch a packaging instance and a validating instance on Amazon EC2 and then create and validate a new package. For more information, see the following topics:

- [Packaging an Application with Amazon WAM \(p. 17\)](#)
- [Packaging and Validating Your Applications \(p. 17\)](#)

To create a package

To create a new package, use Windows Remote Desktop to connect to your package instance and then use Amazon WorkSpaces Application Manager Studio to create the package. For more information, see [Connecting to Your Windows Instance Using RDP](#).

1. Start Internet Explorer and download the Windows installer for Chrome browser at [Alternate \(offline\) Google Chrome installer \(Windows\)](#). Download the installer for all user accounts.
2. Double-click the desktop shortcut to start Amazon WorkSpaces Application Manager Studio.
3. On the Amazon WorkSpaces Application Manager Studio Dashboard, choose **New**.

**Amazon WorkSpaces Application Manager
Administrator's Guide
Packaging Specific Applications**

4. In **1. Application settings**, do the following:
 - For **Title**, enter the name of the application. This text appears on the **Dashboard** tab and in the Amazon WorkSpaces Application Manager Studio to identify this application.

For **Description**, enter a short description for the application.
5. In **2. Application installation**, select an operating system and choose **Install**.
6. In the **Capture Application Installation** dialog box, do the following:
 - For **Installer path**, enter the path to the Chrome installer.
 - For **Capture file and registry activity for the following processes**, select **Installer process and sub-processes only**.
 - Select **Ignore changes under the Installer path specified above**.
 - For **After capturing is complete, do the following actions during the merge**, select **Replace short path names with long ones in the registry**.
 - Choose **Launch**.

The installer for Chrome starts after you choose **Launch**.

7. Skip **3. Optional settings**.
8. Choose the **Registry** tab and then do the following:
 - Open the context (right-click) menu for **<ProgramFilesX86>\Google\Chrome\Application\chrome.exe** and then choose **Properties**.
 - In the **Properties** dialog box, select **Installed-temporary (layer 2)** under **Disposition** and then choose **OK**.
 - Open the context (right-click) menu for **HKEY_LOCAL_MACHINE\Software** and then choose **Add Key**.
 - In the **Add Registry Key** dialog box, select **Create from this existing registry key**, and then choose **Browse**.
 - In the **Browse System Registry** dialog box, choose **HKEY_LOCAL_MACHINE\Software\Policies** and then choose **OK**.
 - Choose **OK** to close the **Add Registry Key** dialog box.
 - Open the context (right-click) menu for **HKEY_LOCAL_MACHINE\Software\Policies**, and then choose **Add Key**.
 - In the **Add Registry Key** dialog box, select **Create a new key with the following name**, type **Google**, and then click **OK**.
 - Repeat the previous two steps to create the **HKEY_LOCAL_MACHINE\Software\Policies\Google\Update** registry key.
 - Open the context (right-click) menu for **HKEY_LOCAL_MACHINE\Software\Policies\Google\Update**, and then choose **Add Value**.
 - For **Name** in the **Add Registry Value** dialog box, type **AutoUpdateCheckPeriodMinutes**. For **Type**, select **REG_DWORD**. For **Data**, type **0**, and then choose **OK**.
9. Choose the **Project** tab and then do the following in **4. Package build**:
 - For **Label**, enter a label for the package. This label appears in the Amazon WAM console as the version label.
 - For **Command line**, type **"Start the Chrome browser from the Desktop shortcut, the Start menu, or the Quick Launch toolbar."**. Select **Set the working folder to**

the folder of the command line executable, if any, select **Set the icon to the source file of the command line executable, if any**, and then choose **OK**.

10. Choose **Save** on the toolbar to save the application package project.
11. In **5. Package upload**, choose **Upload** and then **Close**.

Packaging Mozilla Firefox with Amazon WAM

To create a package, launch a packaging instance and a validating instance on Amazon EC2 and then create and validate a new package. For more information, see the following topics:

- [Packaging an Application with Amazon WAM \(p. 17\)](#)
- [Packaging and Validating Your Applications \(p. 17\)](#)

Important

This procedure shows how to disable automatic updates. You will need to manually update the version. To create a version update, see [Updating an Application \(p. 12\)](#).

To create a package

To create a new package, use Windows Remote Desktop to connect to your package instance and then use Amazon WorkSpaces Application Manager Studio to create the package. For more information, see [Connecting to Your Windows Instance Using RDP](#).

1. Start Internet Explorer and download the Windows installer for Firefox Extended Support Release at [Download Firefox Extended Support Release in your language](#).
2. Double-click the desktop shortcut to start Amazon WorkSpaces Application Manager Studio.
3. On the Amazon WorkSpaces Application Manager Studio Dashboard, choose **New**.
4. In **1. Application settings**, do the following:
 - For **Title**, enter the name of the application. This text appears on the **Dashboard** tab and in the Amazon WorkSpaces Application Manager Studio to identify this application.
 - For **Description**, enter a short description for the application.
5. In **2. Application installation**, select an operating system and choose **Install**.
6. In the **Capture Application Installation** dialog box, do the following:
 - For **Installer path**, enter the path to the Firefox installer.
 - For **Command line parameter**, type `-ms` to do a silent install.
 - For **Capture file and registry activity for the following processes**, select **Installer process and sub-processes only**.
 - Select **Ignore changes under the Installer path specified above**.
 - For **After capturing is complete, do the following actions during the merge**, select **Replace short path names with long ones in the registry**.
 - Choose **Launch**.

The installer for Firefox starts after you choose **Launch**.

7. On the desktop of the packaging instance, start Notepad, and then do the following to create a configuration file that disables the auto-update feature and sets the homepage and a script file to use the configuration file:

**Amazon WorkSpaces Application Manager
Administrator's Guide
Packaging Specific Applications**

- Copy the following code into Notepad and save the file as C:\Program Files (x86)\Mozilla Firefox\mozilla.cfg:

```
// Disable automatic updates
pref("app.update.enabled", false);
pref("app.update.auto", false );
pref("app.update.mode", 0);
pref("app.update.service.enabled", false);

// Disable the default browser check
pref("browser.shell.checkDefaultBrowser", false);
pref("browser.startup.homepage_override.mstone", "ignore");

// Set the homepage
pref("browser.startup.homepage", "https://www.amazon.com");
```

- Copy the following code into Notepad and save the file as C:\Program Files (x86)\Mozilla Firefox\defaults\pref\local-settings.js:

```
//
pref("general.config.obscure_value", 0);
pref("general.config.filename", "mozilla.cfg");
```

Note

The first line in the configuration and files must be two forward slashes (//).

8. Choose the **Files** tab, and then do the following: Open the context (right-click) menu for **<ProgramFilesX86>\Mozilla Firefox** and then choose **Add File**. In **Source file name** on the Add File dialog box, enter **C:\Program Files (x86)\Mozilla Firefox\mozilla.cfg**, and then choose **OK**.
 - Open the context (right-click) menu for **<ProgramFilesX86>\Mozilla Firefox** and then choose **Add File**. In **Source file name** on the Add File dialog box, enter **C:\Program Files (x86)\Mozilla Firefox\mozilla.cfg**, and then choose **OK**.
 - Open the context (right-click) menu for **<ProgramFilesX86>\Mozilla Firefox\defaults\pref** and then choose **Add File**. In **Source file name** on the Add File dialog box, enter **C:\Program Files (x86)\Mozilla Firefox\defaults\pref\local-settings.js**, and then choose **OK**.
9. Choose the **Project** tab and then do the following in **4. Package build**:
 - For **Label**, enter a label for the package. This label appears in the Amazon WAM console as the version label.
 - For **Command line**, choose **Launch Settings**. In the **Application Launch Settings** dialog box, enter **C:\Program Files (x86)\Mozilla Firefox\firefox.exe**. Select **Set the working folder to the folder of the command line executable, if any**, select **Set the icon to the source file of the command line executable, if any**, and then choose **OK**.
10. Choose **Save** on the toolbar to save the application package project.
11. In **5. Package upload**, choose **Upload** and then **Close**.

Validating an Application Package with Amazon WAM

An application package is validated on a special EC2 instance. The first time you access the Amazon WorkSpaces Application Manager console, an Amazon EC2 AMI is shared with you. Use this AMI to launch the validation instance.

Topics

- [Launching the Validation Instance \(p. 30\)](#)
- [Validating the Application Package \(p. 30\)](#)

Launching the Validation Instance

We recommend that you launch the validation instance into its own VPC to make launching and configuration easier. You can, and should, use the same VPC that you used for the packaging instance. For more information, see [Launching the Packaging Instance \(p. 18\)](#).

Before launching the validation instance, you must create the `AmazonWamAppPackaging` IAM role. For more information, see [Application Packaging Role \(p. 3\)](#).

To launch a validation instance, follow the steps in the [Launching Your Instance from an AMI](#) topic in the *Amazon EC2 User Guide for Microsoft Windows Instances*. The following are specific settings for launching a validation instance:

- In step 2, select the region in which your VPC is located.
- In step 4, select the **My AMIs** tab, and select **Shared with me**, and select the **Amazon WAM Admin Player AMI**.
- In step 5, for best performance, we recommend **m3.xlarge** or better for the instance type. Normal Amazon EC2 instance charges apply.
- In step 6, on the **Configure Instance Details** page, enter the following:
 - Under **Network**, select your VPC.
 - Under **Subnet**, select the public subnet in the VPC.
 - Under **IAM role**, select the **AmazonWamAppPackaging** role, which you created in [Application Packaging Role \(p. 3\)](#).
- In step 9, on the **Configure Security Group** page, select a security group that allows access to the instance over port 3389 (RDP) from your network.

Validating the Application Package

After the validation instance is launched, use it to validate your pending application packages. AWS recommends that you launch an entirely new validation instance for each application package that is created.

To validate an application package

1. Using a Remote Desktop client, connect to the validation instance. The remaining instructions are performed on the validation instance.
2. Launch the Amazon WAM Admin Player and select **Pending Apps** in the navigation bar. Any applications that need to be validated are displayed.

3. Select the application package to be validated. If the package is a new package, click **Install**. If the package is a new version of an existing package that is already installed on the instance, click **Upgrade**. The application is installed or upgraded on the validation instance.
4. Launch the application as your users will, and verify that it works correctly.

If the application package does not install or run as expected, you can update the package by creating a package update and validating the update. For more information, see [Creating an Application Update](#) (p. 24).

5. Click **Approve** to mark the package as testing completed.

Using the application package, you can now create a new application, or create a new version of an existing application. For more information about creating a new application, see [Uploading Your Own Applications](#) (p. 8). For more information about creating a new version of an existing application, see [Create a New Version](#) (p. 13).

Troubleshooting Amazon WAM Packaging Issues

The following are some common problems that you may run into when packaging an application.

Topics

- [Application Packaging Issues](#) (p. 31)
- [Device Driver and Service Issues](#) (p. 34)
- [I am missing shortcuts, files, and folders.](#) (p. 37)
- [Other Issues](#) (p. 38)
- [User Account Issues](#) (p. 39)

Application Packaging Issues

Topics

- [My anti-virus and firewall applications do not work in Amazon WAM Admin Player.](#) (p. 31)
- [My application does not run because of a licensing issue.](#) (p. 32)
- [My application fails to start.](#) (p. 32)
- [My application is missing dependencies.](#) (p. 32)
- [File Associations Do Not Work](#) (p. 33)
- [How can I hide the Add/Remove Programs entry for my application?](#) (p. 33)
- [I cannot print from my application.](#) (p. 34)
- [I cannot retrieve my application list.](#) (p. 34)

My anti-virus and firewall applications do not work in Amazon WAM Admin Player.

While you can package various anti-virus and firewall applications, these applications may include drivers that may limit Amazon WAM Admin Player. For example, a firewall application may attempt to block Amazon WAM Admin Player from properly fetching data over the Internet by default.

Add Amazon WAM components to any exceptions rules when creating the package (for example, add `streamingcore.exe` to a firewall allow list).

My application does not run because of a licensing issue.

Many applications employ some form of copy protection to prevent theft and enforce proper licensing. Amazon WorkSpaces Application Manager Studio does not remove copy protection. If you need to enable multiple users to use an application, package the application so that it uses a serial number for high volume licensing so that multiple systems can activate with a single key (for example, Microsoft products use Volume License Keys). Otherwise, the application prompts users for a license and not all systems will be able activate and make the application fully functional.

Test the package on other systems to ensure that there is no form of license enforcement that you are not already aware of. If the package does not work or prompts for licensing, retest the package on the packaging instance. If the application binds to hardware, it should work on the same system. For information about supported licensing models, contact the software ISV.

My application fails to start.

Sometimes an application can fail to start when the command line is incorrect. You might have to specify a different executable at the command line or you might need to add parameters. Check the original application shortcut, typically found on the **Start** menu, and use the same command line and parameters as what appears in **4. Package build** of the Amazon WorkSpaces Application Manager Studio.

Another problem may be that some applications modify the `PATH` environment variable of the local machine during installation or after starting for the first time. There may be some system-specific entries included in the `PATH` variable (for example: `PATH = %PATH%;C:\ApplicationPath1;C:\ApplicationPath2`). Ensure that the package does not contain system-specific entries by choosing the **Environment** tab, choosing **Path**, and checking for the any application-specific path environment variables in the **Value** field.

My application is missing dependencies.

Many applications require the presence of a specific runtime library or a minimum version of the .NET Framework. Amazon WorkSpaces Application Manager Studio comes with the latest builds of the following C++ runtime libraries:

- Visual C++ 2008 x86 and x64
- Visual C++ 2005 x86 and x64

If your application requires other dependencies, install the dependencies before you install the application. Set the default disposition of the dependency files and registry keys to Virtual-isolated (layer 4) before you upload your package.

To set the default disposition of files and registry keys

1. In Amazon WorkSpaces Application Manager Studio, choose the **Files** tab on the top of the project pane.
2. Open the context menu (right-click) for the dependency file or folder and choose **Properties**.
3. In the **Properties** dialog box, choose **Virtual-isolated (layer 4)**, and then choose **OK**.
4. Select the **Registry** tab.
5. Open the context menu (right-click) for the registry hive or key and choose **Disposition**.
6. In the **Registry Key Disposition** dialog box, select **Virtual-isolated (layer 4)**, and then choose **OK**.

File Associations Do Not Work

Windows manages file extension associations to applications under the `HKEY_CLASSES_ROOT` hive of the Windows registry. The file type must be set in the registry and the application executable must not be hidden. Make sure that the file extensions you need exist in the **Registry** tab of the project. If the extension is missing, manually add the extensions. You can also package a new version of the application where you select **All processes** under **Capture file and registry activity for the following processes** in the **Capture Application Installation** dialog box, run the application, and configure the default file associations. Check that the application executable file is not hidden and the disposition is set to Virtual-integrated (layer 3).

To create a new package version

To create a new package version, you need the installation files.

1. In Amazon WorkSpaces Application Manager Studio, choose the **Dashboard** tab on the top of the project pane, choose your project in the list, choose **Extract** in the toolbar, and then choose **Close**.
2. Under **2. Application installation**, choose **Install**.
3. On the **Capture Application Installation** dialog box, do the following:
 - In **Installer path**, enter the path and file name to your installer.
 - In **Command line parameter**, enter any parameters required to install your application.
 - Under **Capture file and registry activity for the following processes**, select **All processes**.
 - Select **Ignore changes under the Installer path specified above**.
 - Choose **Launch**, install your application, choose **OK**, and then choose **Close**.
4. Choose the **Files** tab.
5. Open the context (right-click) menu for your executable file and then choose **Properties**.
6. In **File Properties**, under **Disposition** select **Virtual-integrated (layer 3)**, and then choose **OK**.
7. Choose the **Project** tab and then do the following:
 - In **4. Package build**, type text to describe this new version in **Label**. This text appears as a version of the package.
 - In **5. Package upload**, choose **Upload**.

How can I hide the Add/Remove Programs entry for my application?

Many application add an entry in the **Add or Remove Programs or Programs and Features** list in the Windows Control Panel. This entry can be confusing because the application is virtual and cannot be uninstalled. You can remove this entry from the list.

To hide the application entry

1. In the project in Amazon WorkSpaces Application Manager Studio, choose the **Registry** tab.
2. Open the context (right-click) menu of the following registry key and choose **Delete**.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
```

I cannot print from my application.

Some application install new printer devices, which can be included in a package. On the **Files** tab of Amazon WorkSpaces Application Manager Studio, the **spool** folder is a subdirectory found under the **SystemDir** system folder. Applications that come with printer drivers may not work properly if this folder is removed from the **Files** tab. On the **Registry** tab, the following registry key contains references to the printer drivers in the `spool` folder:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Environments\Windows  
NT x86\Drivers
```

If a new printer does not appear in the application, do the following:

1. In Amazon WorkSpaces Application Manager Studio, choose the **Project** tab on the top of the project pane.
2. Under **3. Optional settings**, choose **Settings**.
3. In the **Settings** dialog box, select **Virtualization**. Under **Special requirements**, select **Restart Print Spooler** and then choose **OK**.

I cannot retrieve my application list.

If you receive an error from the Amazon WorkSpaces Application Manager Studio stating that your app list could not be retrieved, verify that you have created the proper IAM role as explained in [Application Packaging Role \(p. 3\)](#), and that you applied that role to the packaging instance.

Device Driver and Service Issues

Topics

- [How do I package drivers for Plug and Play devices? \(p. 34\)](#)
- [A service shows as running and cannot be stopped. \(p. 35\)](#)
- [Windows services or drivers do not start. \(p. 36\)](#)

How do I package drivers for Plug and Play devices?

Plug and play drivers for devices such as printers are installed in two phases:

- Amazon WorkSpaces Application Manager Studio uses the installer to copy the driver files to the hard drive.
- Windows associates the files with the correct device.

Windows Server 2008 R2 uses a protected repository for trusted device drivers called the Driver Store. To set up a device drive, use a configurable AppEvent that uses the `pnputil.exe` utility. This utility adds and removes drivers from the Driver Store. The syntax for adding drivers using the utility is:

```
pnputil.exe -i -a "C:\<folder>\driver.inf"
```

Caution

Do not attempt to set files in the Driver Store to Installed-permanent (layer 1) or Installed-temporary (layer 2) because it causes virtualization to fail.

To exclude device driver files

1. In Amazon WorkSpaces Application Manager Studio, choose the **Files** tab on the top of the project pane.
2. Open the context menu (right-click) for the `<WindowsDir>\inf\` folder and select **Exclude**.
3. To exclude all virtual files used by the device driver, do the following:
 - On the instance Desktop, choose **Start**, type `devmgmt.msc`, and then press Enter to open Device Manager.
 - Double-click the device driver in the list to open the **Properties** dialog box.
 - In **Properties**, choose the **Driver** tab, choose **Driver Details**, make note of the files listed in the **Driver File Details** dialog box, and then choose **OK**.
 - In Amazon WorkSpaces Application Manager Studio, choose the **Files** tab on the top of the project pane.
 - Open the context menu (right-click) for each file and select **Exclude**.
4. To exclude files for TWAIN devices, do the following:
 - In Amazon WorkSpaces Application Manager Studio, choose the **Files** tab on the top of the project pane.
 - Open the context menu (right-click) for each driver file under `<WindowsDir>\SSDriver` and choose **Properties**.
 - In the **Properties** dialog box, choose **Installed-temporary (layer 2)** under **Disposition**, and then choose **OK**.

To create the configurable AppEvent

1. In Amazon WorkSpaces Application Manager Studio, choose the **Projects** tab on the top of the project pane, and then under **3. Optional settings**, choose **Settings**.
2. In **Settings**, select **Configurable AppEvents**, and then choose **Add**.
3. In the **Configurable AppEvent Settings** dialog box, select **After activation** under **Trigger**.
4. • In **File name in Handler command line** type `%SystemRoot%\system32\pnputil.exe`. Under **Handler Launching**, select **Run elevated (only supported on Windows Vista or later)** and then choose **OK**.

Note

Device drivers remain on the system even after the application has been removed by Amazon WAM Admin Player. This is expected behavior.

A service shows as running and cannot be stopped.

Windows Task Manager tracks all executable files and services with Virtual-integrated (layer 3) or Virtual-isolated (layer 4) disposition. When the executable file or service is in use, Windows Task manager shows this file with a **Status of Running**.

To avoid usage tracking, set the disposition of these files (such as Service ImagePath) and services to Installed-temporary (layer 2) disposition.

In addition, some services may be configured to restart automatically after an interruption.

To configure an interrupted service to restart automatically

1. In Amazon WorkSpaces Application Manager Studio, choose the **Registry** tab on the top of the project pane.
2. Remove the Failure action key from the following hive for the service:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
```

Windows services or drivers do not start.

If a service needs to start using a unique user account (this normally defaults to LocalSystem), the application will need a configurable AppEvent to create this account.

Caution

Some systems services have security descriptors that are strictly enforced. When adding a new service, it is important to match the security descriptor of the installed version. Use `sc sdshow <service_name>` at the command prompt to display the security descriptor set for the service. Marking an application to require a system restart after adding a service typically corrects the security descriptors.

If the application has boot or system drivers, mark all of the files and registry keys for those services or drivers as Installed-temporary (layer 2).

To set the disposition of the files and registry keys

1. In Amazon WorkSpaces Application Manager Studio, choose the **Services & Drivers** tab on the top of the project pane.
2. Open the context menu (right-click) for the service or driver.
3. In the **Service Key Disposition** dialog box, select **Installed-temporary (layer 2)**, and then choose **OK**.

Caution

Set both the files and registry keys to Installed-temporary (layer 2) disposition. You may have problems if the files and registry keys do not have the same disposition setting.

4. Choose the **Registry** tab.
5. Open the context menu (right-click) for your service or driver and choose **Exclude**.
6. Choose the **Project** tab.
7. You can choose to allow or avoid a system restart. In **3. Optional settings**, choose **Settings**, and then do one of the following:
 - To require a system restart, select **Require system reboot** under **Special requirements**, and then choose **OK**.
 - To avoid a system restart, do the following to use a configurable AppEvent to start the driver directly, select **Require system reboot** under **Advanced settings**. Select **Configurable AppEvent**, and then choose **Add**. In the **Configurable AppEvent Settings** dialog box, select **After Launch** under **Trigger**. In **File name** under **Handler command line**, type `sc.exe start <driver_name>` and then choose **OK**.

For more information about how services work, go to [How to create a Windows service](#) in the Microsoft Knowledge Base.

Note

A system driver cannot be stopped and removed from a system without a reboot. Amazon WorkSpaces Application Manager Studio removes these files from the system but the driver is still loaded until a reboot occurs.

I am missing shortcuts, files, and folders.

Topics

- [My Shortcut is missing](#) (p. 37)
- [My shortcut is a blank window](#). (p. 37)
- [The text is in the wrong font](#). (p. 37)
- [My folders and files are not in the application in my WorkSpace](#). (p. 38)

My Shortcut is missing

If the shortcut files are missing after the application has been installed, verify the following:

- In order for the shortcut to appear on the **Start** menu, verify that the shortcut is located in the `CommonPrograms` or `UserPrograms` folder. In order for the shortcut to appear on the Windows desktop, verify that the shortcut file is located in the `CommonDesktop` or `UserDesktop` folder.
- Make sure that the **Disposition** for the shortcut files and folders are set to **Virtual-integrated**. This allows that shortcut to be accessible on the client computer's operating system.
- Check that the security setting on the shortcut files do not have **Hide from folder listing** selected. If this option is selected, the shortcut files will not be visible on the client computer.

My shortcut is a blank window.

If the desktop icon for the application is a blank Windows icon, the executable file for the application is probably set to **Prevent Copying** under **Security settings** in the **File Properties** dialog box. This is normal behavior with this setting and the application will function normally.

The text is in the wrong font.

Most applications come with their own program fonts. Fonts will not appear if an application is packaged with **Inject fonts for use only by the application** selected under **Virtualization** in the **Settings** dialog box. Private fonts are only used in the application if it is launched from the Amazon WorkSpaces Application Manager Studio.

To make private fonts available to all applications

1. In Amazon WorkSpaces Application Manager Studio, choose the **Projects** tab on the top of the project pane.
2. Under **3. Optional settings**, choose **Settings**.
3. In the **Settings** dialog box, select **Virtualization**. Under **Fonts**, select **Register fonts for system-wide use** and then choose **OK**.

In addition to the **Fonts** tab, registered fonts can be found in the following locations:

- In the `fonts` directory in `SystemFonts` in the **Files** tab.
- In the following registry key on the **Registry** tab:


```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Fonts
```

My folders and files are not in the application in my Workspace.

If you package an application to an install folder that has the same name as a folder used in a Workspace, the folder in the Workspace may disappear. This can also affect files under common folders marked with Virtual-integrated (layer 3) disposition for integrated, but not merged.

To fix an existing application package with missing folders and files

1. Use a unique name for the install folder.
2. In Amazon WorkSpaces Application Manager Studio, choose the **Files** tab on the top of the project pane.
3. Open the context (right-click) menu and then select **Properties**.
4. In the **File Properties** dialog box, select **Virtual-isolated (layer 4)** under **Disposition** and then choose **OK**.

Other Issues

Topics

- [A warning message about an alternate data stream \(ADS\) appears to my users. \(p. 38\)](#)

A warning message about an alternate data stream (ADS) appears to my users.

Users may receive warning messages about alternate data streams (ADS) when their application starts in WorkSpaces. In most cases, these messages are not a serious problem and can be ignored. In rare cases, the application may require an ADS to use a particular file. In this case, the package should exclude the file from the sandbox. Set the disposition of the file to Installed-permanent (layer 1) or Installed-temporary (layer 2).

To exclude a file from the sandbox and set the disposition

1. In Amazon WorkSpaces Application Manager Studio, choose the **Files** tab on the top of the project pane.
2. Open the context menu (right-click) for the file and choose **Exclude**.
3. Open the context menu (right-click) for the file and choose **Properties**.
4. In the Properties dialog box, select **Installed-permanent (layer 1)** or **Installed-temporary (layer 2)** under **Disposition**, and then choose **OK**.

User Account Issues

My roaming user accounts and folder redirection does not work.

Amazon WorkSpaces Application Manager Studio allows the `HKEY_CURRENT_USER` registry settings and the `%appdata%` folder contents for roaming profile support. Any limitations with Windows roaming profiles still apply.

If you are using folder redirection to a network share through a mapped drive or a UNC path, set the disposition of the folder and the files to the recommended Installed-permanent (layer 1) or Installed-temporary (layer 2). The application will not start if the files or folders are set to Virtual-integrated (layer 3) or Virtual-isolated (layer 4).

For example, if the `%appdata%` folder is being redirected to `\\fileshare\user1\appdata\`, the template folder `?roamingappdata?` must only contain files and folders with Installed-permanent (layer 1) or Installed-temporary (layer 2) disposition. Redirection to another local disk supports files set to the Virtual-integrated (layer 3) or Virtual-isolated (layer 4) disposition.

Caution

With folder redirection, the network folder must exist prior to activating the application. For example, if `%appdata%` is mapped to `n:\{username}\appdata`, the `appdata` folder must already exist on the server.

To enable roaming profile support

1. In Amazon WorkSpaces Application Manager Studio, choose the **Project** tab on the top of the project pane.
2. Under **3. Optional settings**, choose **Settings**.
3. In the **Settings** dialog box, choose **Sandboxing**, choose **Allow Windows to roam application settings** under **Application settings**, and then choose **OK**.

Unique User Accounts

Some applications create a special user account when you install them. Amazon WorkSpaces Application Manager Studio cannot capture user accounts because they are not transferable between WorkSpaces. You need to create a configurable AppEvent to programmatically replicate the user accounts and password on the host computer when Amazon WorkSpaces Application Manager Studio packages the application.

Note

User accounts may appear as a security identifier (SID) in the registry. For example, S-1-5-18 is the local system account. For more information, go to [Well-known security identifiers in Windows operating systems](#) in the Microsoft Knowledge Base.

Amazon WorkSpaces Application Manager Limits

The following table lists the limits for Amazon WorkSpaces Application Manager (Amazon WAM). Unless indicated otherwise, these limits are per region.

Amazon WAM Limits

Resource	Limit	Comments
Application assignments	50 per user	
Packages	None	There is no limit to the number of applications you can package, but storage fees will be applied if your packages exceed 100GB.

Document History

The following table describes important additions to the *Amazon WAM Administration Guide*. We also update the documentation frequently to address the feedback that you send us.

- **Latest documentation update:** April 9th, 2015

Change	Description	Date Changed
Initial release	Initial release of the <i>Amazon WAM Administration Guide</i> .	April 9th, 2015