SAS-wsm Configuration Guide

v. 1.0.0, 20 July 2022

Contents

Obtain the SAS-wsm Utility 1

Understanding Your Deployment 2

- Locate Current SAS Services 2
- Determine Correct Startup Order of SAS Services 3
- Locate the SAS Configuration Directory 3

Setting the Initial Configuration 4

- Define the Start-up Order 4
- Define the Stop Order 5
- Define the SAS Configuration Directory Path 5
- Save the Updated SAS-wsm.ps1 5

Testing Your SAS-wsm Configuration 6

- Testing Considerations 6

Running SAS-wsm 6

- Start Services 6
- Stop Services 6
- Check Current Status of Services 6

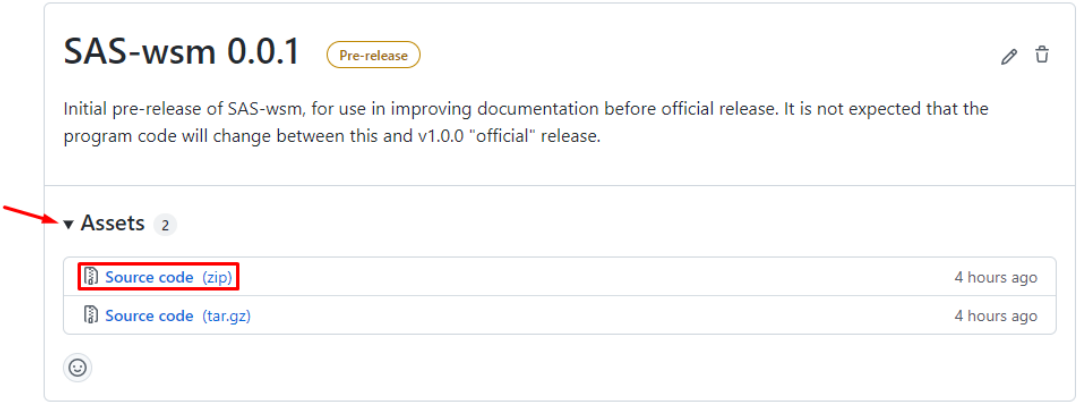
Troubleshooting SAS-wsm 7

- Known Issues 7

Obtain the SAS-wsm Utility

If you have not already done so, download the latest version of SAS-wsm from the GitHub project page, and save SAS-wsm to the local machine where you control your services:

1. Navigate to https://github.com/sassoftware/sas-wsm/releases .
2. Click the arrow next to **Assets** under the most-recent release, near the top of the page.
3. Click the link to download the source code in either .zip or .tar.gz format.



The ZIP format is recommended for Windows.

4. Move the downloaded file to the location from which you would like to run SAS-wsm.
5. Extract the downloaded files using your preferred utility.

Understanding Your Deployment

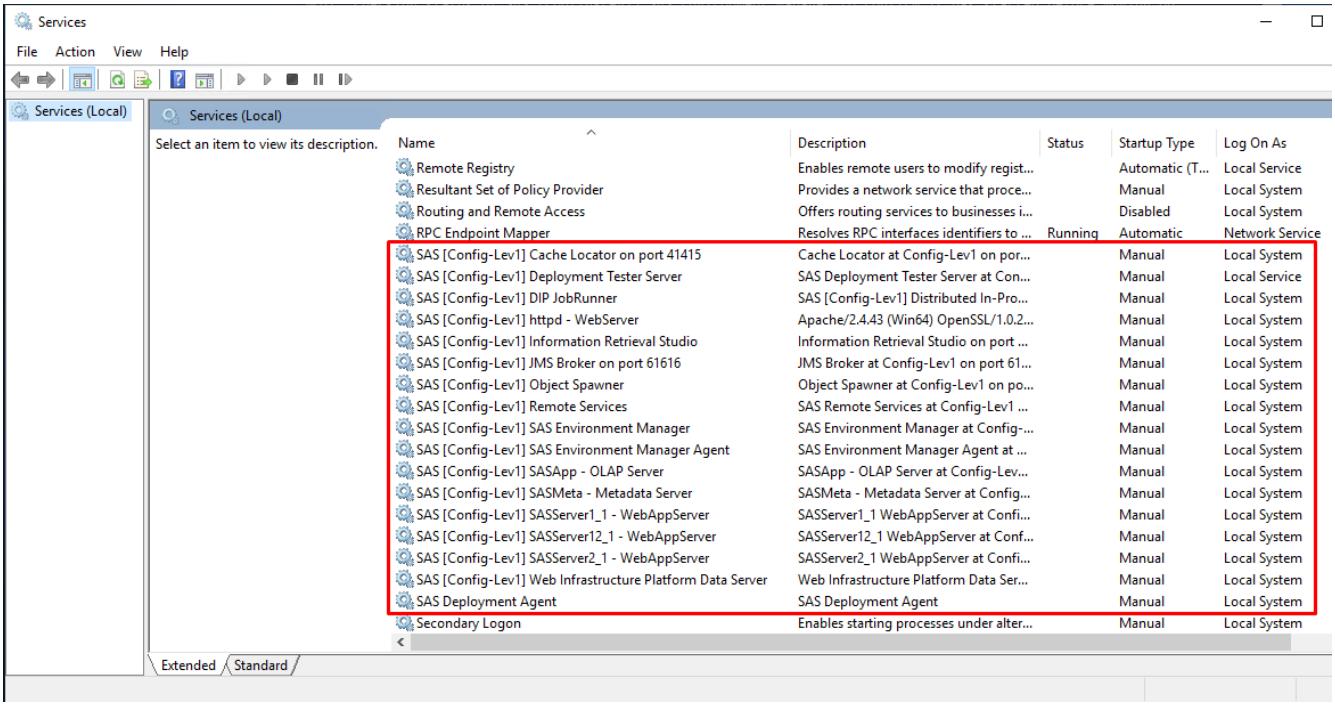
In order to correctly configure SAS-wsm, you must understand your current SAS deployment. This section provides guidance about how to locate current SAS services and determine their correct startup order.

Locate Current SAS Services

To view the SAS services that are deployed on your machine, open the Windows Services manager.

- Open the start menu (or press Windows+R) and type **services.msc**, then press **Enter**.

Scroll down the list of Windows Services until you find the section of services beginning with **SAS**.



Determine Correct Startup Order of SAS Services

Once you have located the SAS services that are deployed on the machine, it is critical that you determine the correct order with which to start those services. Some SAS services have dependencies on other services being available, which might result in start-up errors if those dependencies are not running when a later service is started.

If you have ever restarted SAS services on your machine without a full machine reboot, you have likely already determined the startup order. If so, you should use that documented start-up order for configuring SAS-wsm.

If you do not know the correct start-up order, reference the Overview of Server Operation documentation.

The following are the typical "critical" SAS services, which are usually started in the following order:

1. SAS Metadata Server
2. SAS Web Infrastructure Platform Data Server
3. Additional SAS Data Servers, if configured
4. SAS Object Spawner
5. SAS Cache Locator
6. SAS JMS Broker
7. SAS Web Server (HTTPD)
8. SAS Web Application Server SASServer1_1
9. Additional SAS Web Application Servers, if configured (SASServer2_1, SASServer12_1, and so on)

Most of the other SAS services can be started at any time, with the recommended order being as follows:

- Start any additional metadata-tier resources after the Metadata Server.
- Start compute-tier services after the Object Spawner.
- Start MidTier (and all other) services after the Web Application Servers.

Make note of your correct service start-up order so that you can use it in the next section when you configure SAS-wsm for your environment.

If you are unsure about when to start a SAS service, contact SAS Technical Support for assistance.

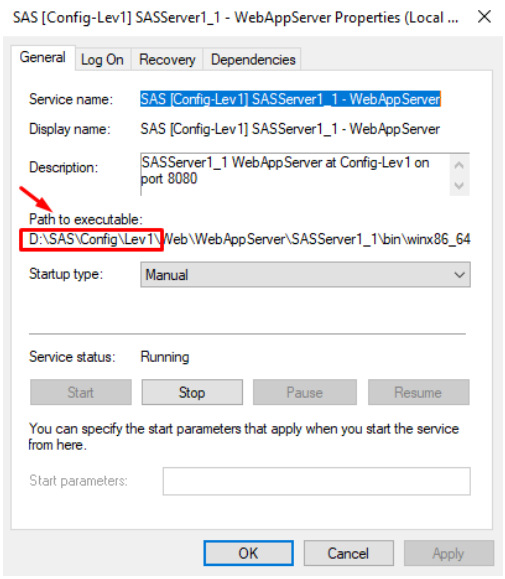
Locate the SAS Configuration Directory

When SAS software is deployed in a Planned Deployment, a SAS Configuration Directory is created to store configuration information about various SAS components. SAS-wsm must know where this directory is located in order to reference component log files and verify start-up states.

If you are not sure where this location is, open the properties of a SAS Web Application Server from **services.msc**.

In the **Path to executable** property, you should see a file path similar to that shown in the example to the right, containing a folder named Lev*N* (where *N* is a number). This Lev*N* folder is the root of your SAS Configuration Directory.

Copy or make note of this path because you need to reference it in the next section.



Setting the Initial Configuration

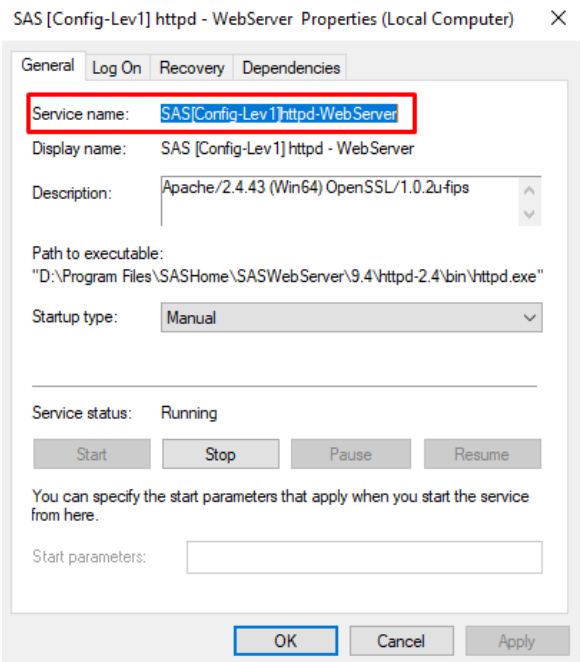
Inside the extracted SAS-wsm utility files that you previously downloaded, open the file SAS-wsm.ps1 in your preferred text editor. You must modify this file for your specific SAS deployment.

Define the Start-up Order

Using the correct SAS services start-up order that you previously determined, add them to the SAS-wsm.ps1 file under the function **Start-SAS**. You should see a comment here stating **#DEFINE-VARIABLE** to show where these edits should be made, followed by an example service start-up order.

Delete this comment and the example start-up order, and replace them with the correct start-up order for your deployment, using the service names as shown in **services.msc**.

For accuracy, it is recommended that you copy the service names from services.msc by right-clicking the name of the service, selecting **Properties**, and copying/pasting the name from the **Service Name** field:



Be sure to use the **Service Name** field and ***not*** the **Display Name** field. These values might vary for some services, such as the httpd Web Server, shown above. Windows fails to locate a service by its Display Name.

The following formatting should be followed when you define services:

- Place the name of the service inside double quotation marks (").
- End each service definition with a comma (,) unless it is the last service that you are defining.

Define the Stop Order

Using the start-up order that you defined in the previous step, you can now define the stop order. Copy the start-up order and place it in the function **Stop-SAS** of the SAS-wsm.ps1 file.

Similar to the previous step, you should see a comment here stating **#DEFINE-VARIABLE** to show where these edits should be made. Also shown is an example service stop order. Delete this comment and the example stop order.

In most cases, **the stop order for SAS services should be the opposite of the start order**. Move the last service that you start to the top of the stop list, repeating until all services have been shuffled to their opposite positions. If done correctly, the last service that is stopped is likely the SAS Metadata Server.

Be sure to verify that service definitions have been formatted correctly following your re-ordering of the services in this section. As a reminder, the following formatting should be followed when defining services:

- Place the name of the service inside double quotation marks (").
- End each service definition with a comma (,) unless it is the last service that you are defining.

Define the SAS Configuration Directory Path

SAS-wsm must know where the SAS Configuration Directory is located in order to reference component log files and verify start-up states. Locate the function **Check-WebAppServer-Ready** in the SAS-wsm.ps1 file. Near the top of this function is a variable, *$sasconfigpath*, which defines the SAS Configuration Directory root folder.

Similar to the previous sections, you should see a comment here stating **#DEFINE-VARIABLE** to show where these edits should be made. Also shown is an example SAS Configuration Directory root folder. Delete the example.

Place the correct SAS Configuration Directory root folder path in the variable, ensuring that the formatting is correct. Include the trailing slash (\ ) after the **Lev*N*** folder.

```
149  function Check-WebAppServer-Ready($servicename){ #call function using $_ to send the name of the service we are calling
150
151  $sasconfigpath = "D:\SAS\Config\Lev1\" #DEFINE-VARIABLE: SAS Configuration Path, including Lev#, in quotes. Example: "D:\SAS\Config\Lev1\"
```

Save the Updated SAS-wsm.ps1

With your edits made from the previous sections, be sure to save the SAS-wsm.ps1 file. When SAS-wsm is run to control your SAS services, you call this file.

This might also be a good time to verify that your account has the correct permissions to run a .ps1 file (PowerShell script), because this might be restricted on a per-file level or blocked via GPO. If you do not have the ability to run .ps1 files, discuss the issue with your system administrators and security teams.

Testing Your SAS-wsm Configuration

With the SAS-wsm.ps1 file modified for your specific environment, you should test the utility to verify that it can successfully start, stop, and check the status of your SAS services.

If any errors appear while SAS-wsm is running, there might be a problem with your definitions.

If no errors appear but some services are not controlled as expected, verify that they were defined, and that you copied the correct Service Name from services.msc.

Testing Considerations

A status check can be performed on a running system, and it does not impact the state of SAS services.

A stop/start check causes SAS services to become unavailable while they are stopped. If there are current SAS sessions running, this could cause users to lose their unsaved work. Please ensure that there are no active users on the system before testing a stop and start operation!

For command usage, see the section Running SAS-wsm below.

Running SAS-wsm

To run SAS-wsm, execute the SAS-wsm.ps1 script that you previously modified from a PowerShell session. Provide the script with parameters for the action that you wish to perform (start, stop, status). It is recommended that you also provide a parameter that defines the server name (shortname or fully-qualified domain name [FQDN]). However, this information is not required in the current release.

To open a PowerShell session, open the Start menu, scroll through the list of programs for the folder Windows PowerShell, and select the program named Windows PowerShell under that location.

The example commands shown below assume that you have previously navigated to the location of the SAS-wsm.ps1 file in your PowerShell session. If you have not, reference the full path to SAS-wsm.ps1 in your commands (or use **cd** command to change the current directory of the PowerShell session to where you saved SAS-wsm.ps1).

Start Services

```
.\SAS-wsm.ps1 -servername myserver.sas.com -action start
```

Stop Services

```
.\SAS-wsm.ps1 -servername myserver.sas.com -action stop
```

Check Current Status of Services

```
.\SAS-wsm.ps1 -servername myserver.sas.com -action status
```

Troubleshooting SAS-wsm

We apologize if you encounter issues with SAS-wsm! Please know that while we do our best to test all scenarios before releasing updates, the differences in every individual SAS deployment might result in unexpected behavior. This section will be updated periodically to document known-issues, workarounds, and other potential fixes.

If you encounter problems using SAS-wsm that you are unable to resolve through your own troubleshooting and the information contained in this document, consider letting us know via a GitHub issue on the project or by opening a track with SAS Technical Support.

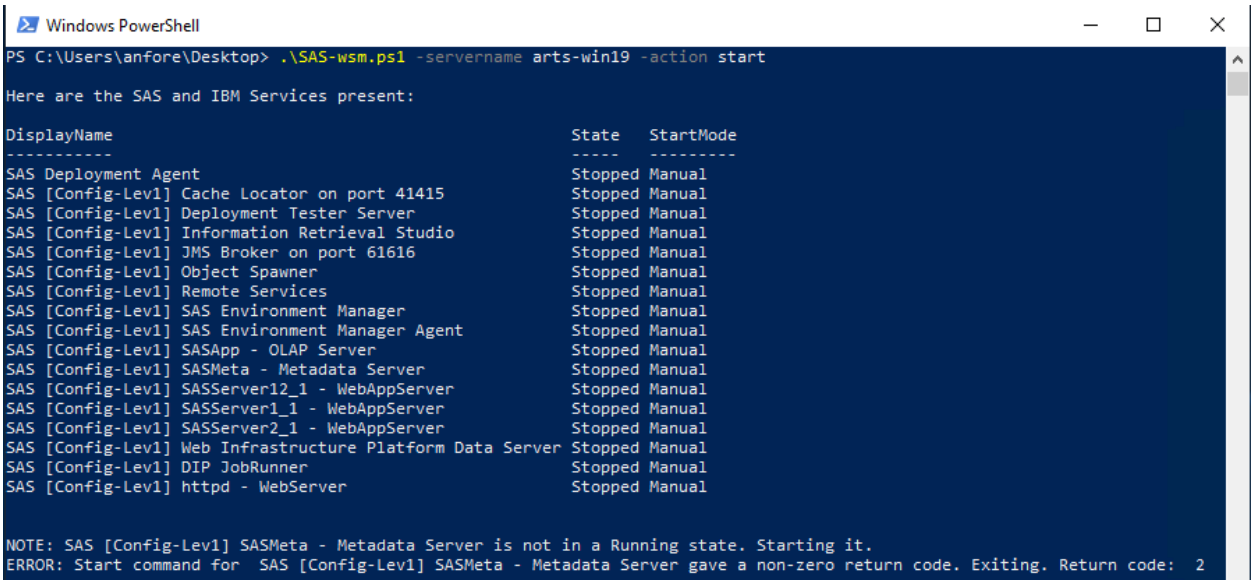
If you encounter a problem but are able to fix it, please consider contributing your fixes to the GitHub project or let us know so that we can resolve the issue in a future release.

When contacting SAS Technical Support about SAS-wsm, please note that our regular support policies might not apply. Refer to the **SUPPORT.md** document included with SAS-wsm for more information.

Known Issues

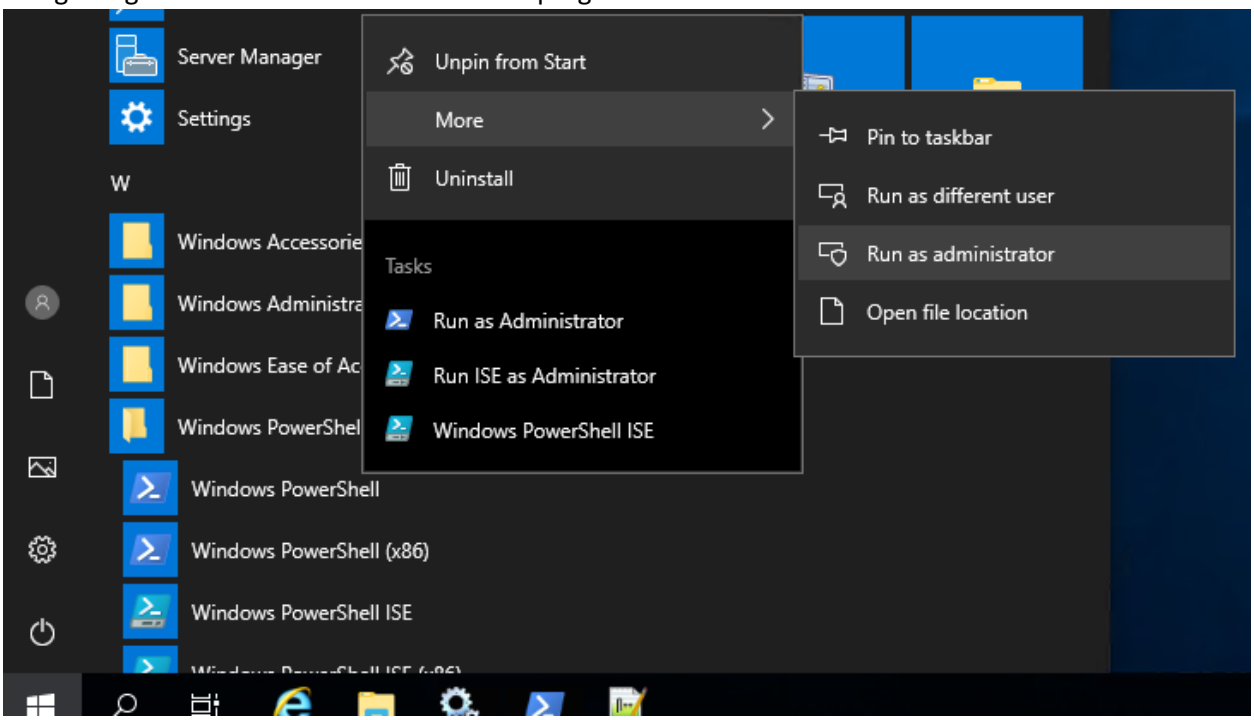
Metadata Server Startup Permissions

When starting services with SAS-wsm, a Windows account that does not own the files under the SAS Configuration Directory might fail to start the Metadata Server. The Metadata Server start-up returns a nonzero exit code, which indicates an error, but no Metadata Server log is produced.



In many environments, the permissions to control SAS services are inherited through an Administrators group that various Windows accounts belong to. To access this group in PowerShell, run your PowerShell session as an administrator. There are many methods to achieve this, with the simplest

being to right-click the Windows PowerShell program and select **Run as Administrator**:



This should enable successful startup: