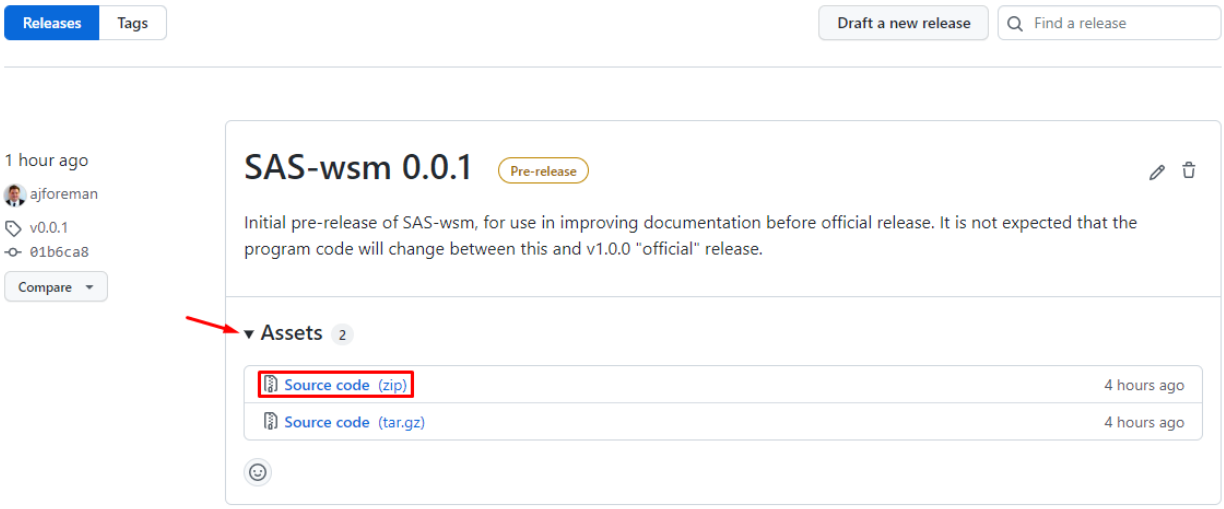
SAS-wsm Configuration Guide
v. 1.1.0, 15 September 2022

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 will be controlling services.

1. Navigate to https://github.com/sassoftware/sas-wsm/releases .
2. Select the drop-down ***Assets*** under the most-recent release, near the top of the page.
3. Download the ***Source Code*** in either .zip or .tar.gz format.
    a. The .zip format is recommended for Windows.



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

Understanding Your Deployment

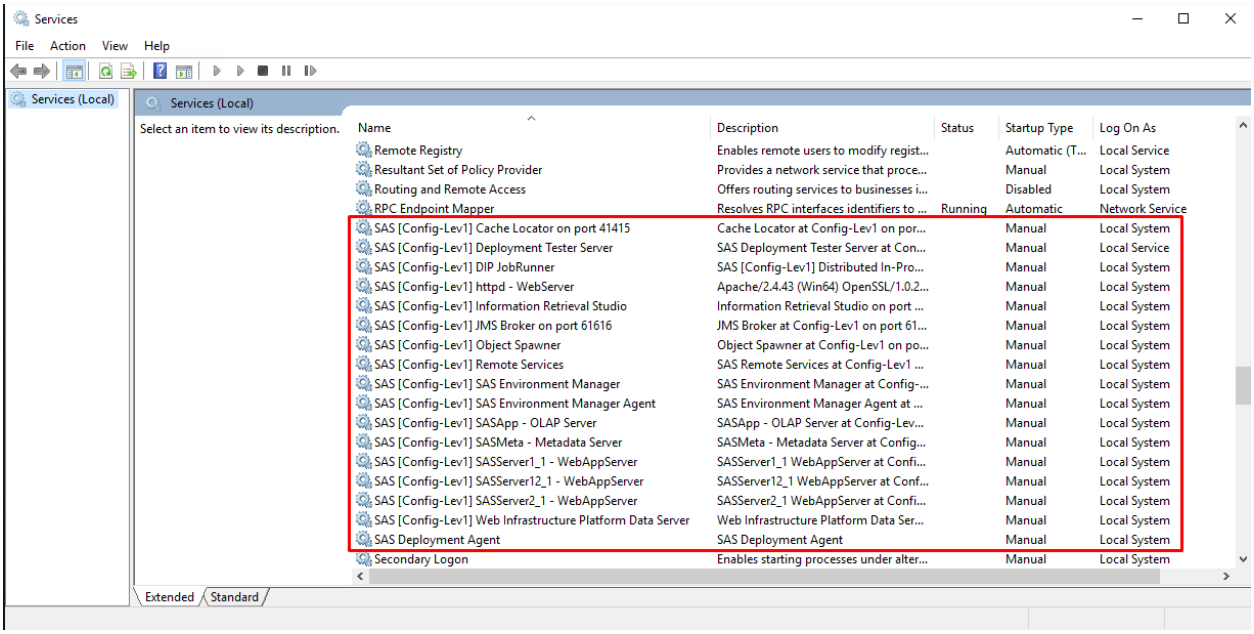
In order to correctly configure SAS-wsm, you must understand your current SAS deployment. This section provides guidance on how to locate current SAS services and their correct startup order.

Locate Current SAS Services

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

- Open the start menu (or press Windows+R) and type **services.msc**, then hit 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 deployed on the machine, it is critical that you determine the correct order to start those services. Some SAS services have dependencies on other services being available, which may result in startup 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, and should use that documented startup order for configuring SAS-wsm.

If you do not know the correct startup order, reference SAS documentation here:
https://go.documentation.sas.com/doc/en/bicdc/9.4/bisag/p0d9d5nzmd8i4yn1usv2l22vpa7t.htm

The following are the typical “critical” SAS services, 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, etc.)

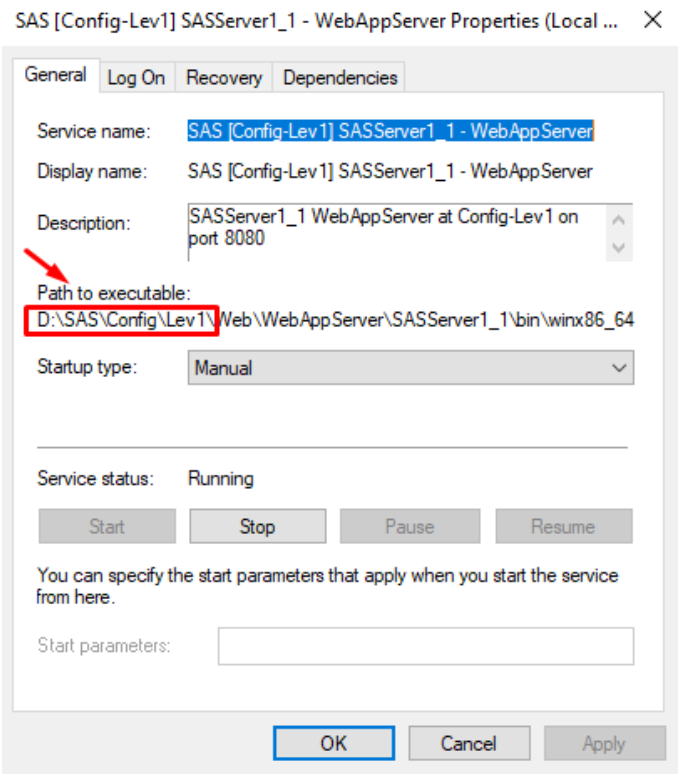
Most other SAS services can be started at any time, with the recommended order being to start any additional Metadata-tier resources after the Metadata Server, Compute-tier services after the Object Spawner, MidTier (and all other) services after the Web Application Servers.

Make note of your correct service startup order so that you can use it in the next section, when we 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 startup states.



If you are not sure of this location, 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 LevN folder is the root of your SAS Configuration Directory.

Copy or make note of this path as we will need to reference it in the next section.

Initial Configuration

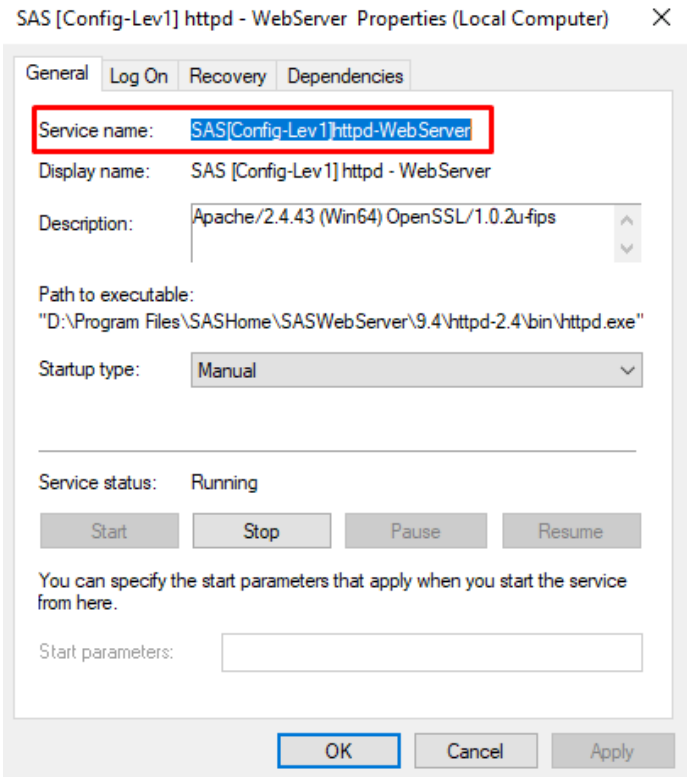
Inside the extracted SAS-wsm utility files you previously downloaded, open the file **example-servers.cfg** in your preferred text editor. The default contents of this file likely incorrect or incomplete for your environment, and are only provided as a reference.

You can modify/delete the current contents of the file, or make a new **.cfg** configuration file for your specific SAS deployment.

Define Startup Order

Using the correct SAS services startup order you previously determined, add them to the **example-servers.cfg** or your new configuration file. Use the service names as shown in **services.msc**.

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



Be sure to use the Service Name field and NOT the Display Name field. These values may vary for some services, such as the httpd Web Server. Windows will fail to locate a service by its Display Name.

The following formatting should be followed when defining services:

- Place the first service to start on line 1, the second service to start on line 2, etc.
- Provide the full service names exactly as seen in services.msc properties, one service per line
- Do not add numeric formatting characters, comments, blank lines, or other superfluous text

Understanding Stop Order

In most cases, **the stop order for SAS services should be the opposite of the start order**.

SAS-wsm will read the defined configuration file backwards (from the last line of the file to the first line of the file) when performing a stop operation. This results in the stop order being the opposite of the start order that was defined.

Define SAS Configuration Directory Path

SAS-wsm must know where the SAS Configuration Directory is located in order to reference component log files and verify startup states. Open the **SAS-wsm.ps1** file in a text editor and locate the section marked **#USER-DEFINED VARIABLES** near the top of the file.

Inside this section is a variable, *$sasconfigpath* , which defines the SAS Configuration Directory root folder. Place the correct SAS Configuration Directory root folder path in the variable, ensuring that the formatting is correct by referencing the example. Include the trailing slash (\ ) after the **Lev*N*** folder.

```
13  ######## USER-DEFINED VARIABLES ########
14
15  # SAS Configuration Directory folder path, including Lev#, in quotes.
16  # $sasconfigpath="C:\SAS\Config\Lev1\"
17  $sasconfigpath = "C:\SAS\Config\Lev1\"
```

Save Updated Configuration and SAS-wsm.ps1 Files

With your edits made from the previous sections, be sure to save your configuration file and the SAS-wsm.ps1 file. When SAS-wsm is run to control your SAS services, you will be calling these files.

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

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 may be a problem with your definitions.

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

Testing Considerations

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

A stop/start check will cause SAS services to become unavailable while they are stopped. If there are current SAS sessions, 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 you previously modified from a PowerShell session. Provide the script with parameters for the action you wish to perform (start, stop, status), and the path to the configuration file you defined deployment services in.

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

The example commands shown below assume 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**). Similarly, the full path to the configuration file must be provided if it is not inside the current working directory, as shown in the 'Start Services' example below.

Start Services

```
.\SAS-wsm.ps1 -action start -cfg "C:\My Folder\docs\example-servers.cfg"
```

Stop Services

```
.\SAS-wsm.ps1 -action stop -cfg example-servers-currentDir.cfg
```

Check Current Status of Services

```
.\SAS-wsm.ps1 -action status -cfg example-servers-currentDir.cfg
```

Troubleshooting SAS-wsm

We apologize if you are encountering 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 may result in unexpected behavior. This section will be periodically updated to document known-issues, workarounds, and other potential fixes.

A recommended first-step in any troubleshooting is to run the built-in validation function. This will help confirm that all expected definitions were set correctly and that the current configuration file is able to be read. To run the validation function, simply call it as you would the operational functions:

```
.\SAS-wsm.ps1 -action validate -cfg "C:\My Folder\docs\example-servers.cfg"
```

If you suspect that you may be defining services that Windows does not identify, you can also run a search on all locatable services which contain the word "SAS" (or "IBM", for Platform LSF) by running:

```
.\SAS-wsm.ps1 -action search
```

If you encounter problems using SAS-wsm which 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 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 may resolve the issue in a future release.

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

Known Issues

Modified server.log Messaging For lastinit Search Based On Hotfix Level

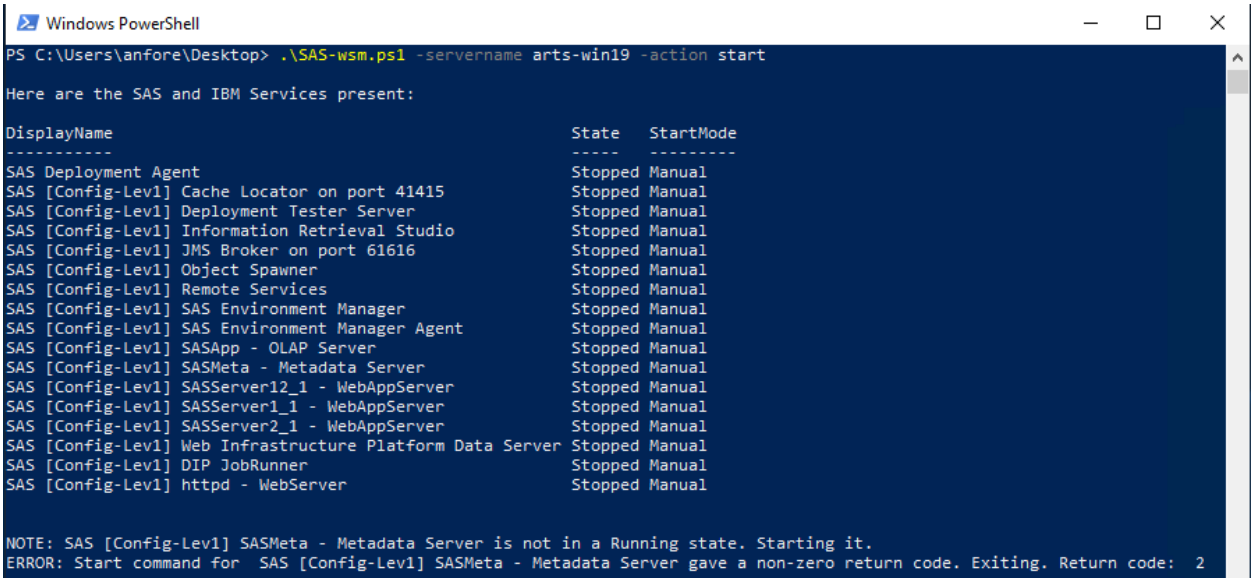
Due to updates in the SAS Web Server and SAS Web Application Server on newer releases of SAS 9.4M6 and SAS 9.4M7, the message printed to a Web Application Server's server.log file when it begins initializing has changed. SAS-wsm is set to use the new messaging by default.

If you are on an older SAS 9.4 maintenance release, have an older weekly revision / hotfix level for SAS 9.4M6/M7, or are told by SAS-wsm that the Web Application Server failed to initialize (on a start operation) but find it to be reported as running afterward, you may need to change the search pattern to use the old messaging syntax.

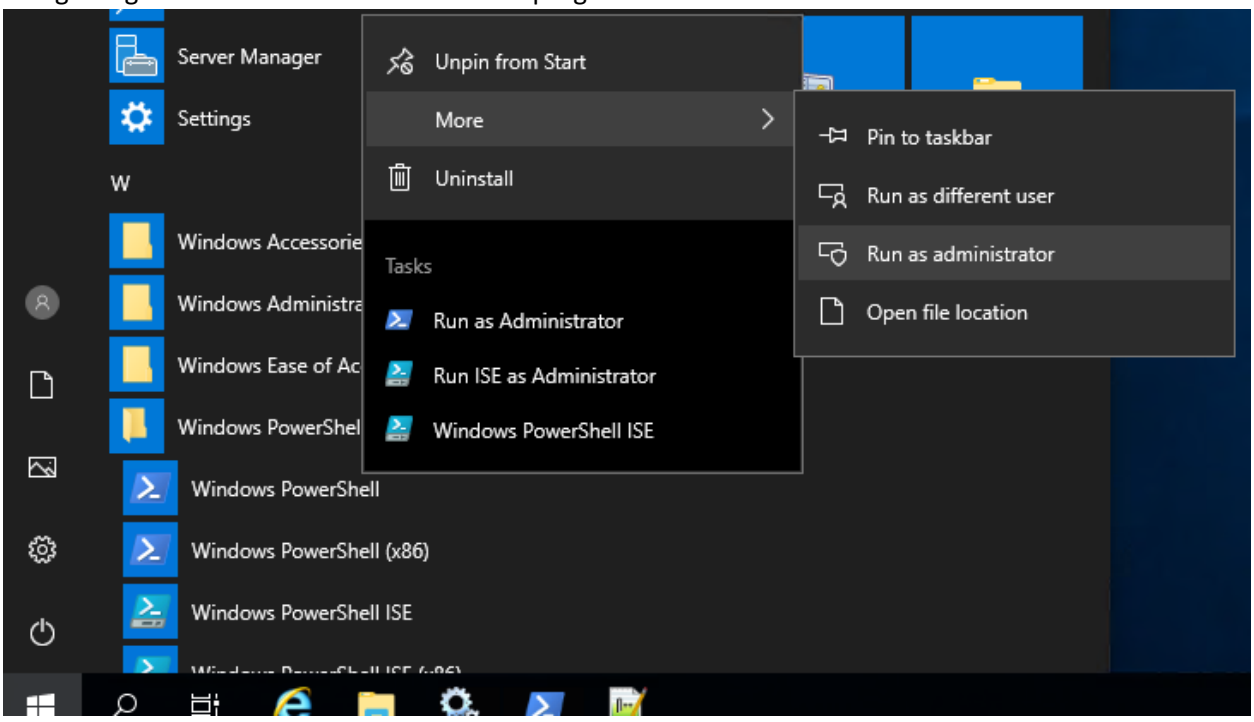
Refer to the comment ***#message changes on hotfix level…*** inside the function named **Check-WebAppServer-Ready** for instructions on how to modify if needed.

Metadata Server Startup Permissions

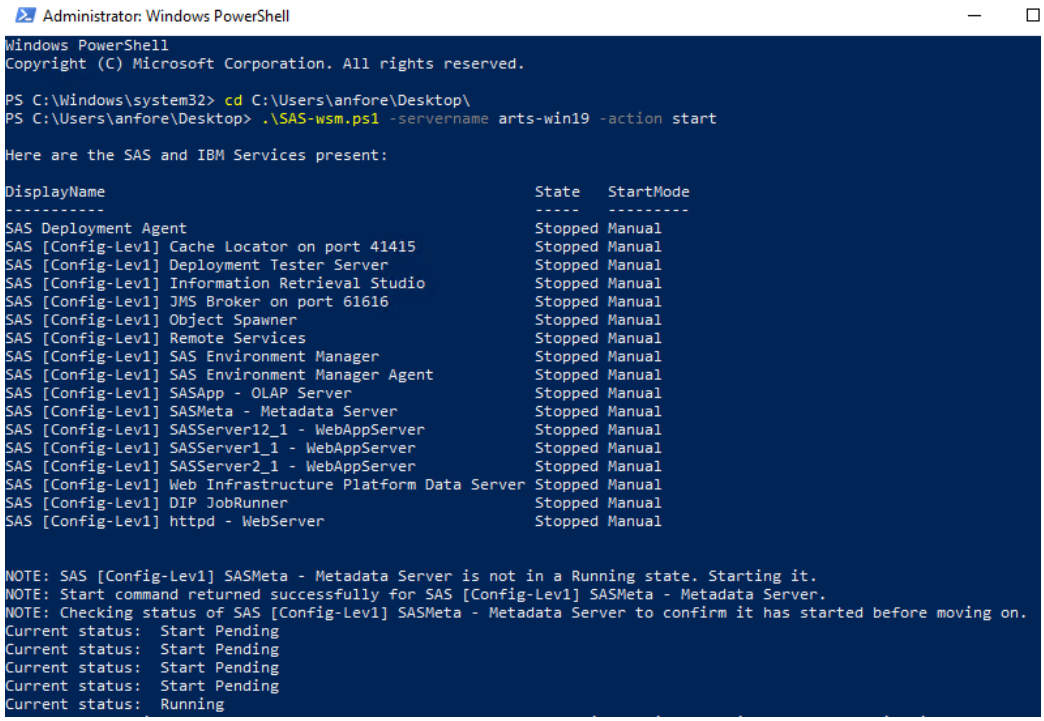
When starting services with SAS-wsm, a Windows account which does not own the files under the SAS Configuration Directory may fail to start the Metadata Server. The Metadata Server startup will return a nonzero exit code, indicating an error, but no Metadata Server log will be produced.



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:


Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Windows\system32> cd C:\Users\anfore\Desktop\
PS C:\Users\anfore\Desktop> .\SAS-wsm.ps1 -servername arts-win19 -action start
Here are the SAS and IBM Services present:
DisplayName State StartMode
SAS Deployment Agent Stopped Manual
SAS [Config-Lev1] Cache Locator on port 41415 Stopped Manual
SAS [Config-Lev1] Deployment Tester Server Stopped Manual
SAS [Config-Lev1] Information Retrieval Studio Stopped Manual
SAS [Config-Lev1] JMS Broker on port 61616 Stopped Manual
SAS [Config-Lev1] Object Spawner Stopped Manual
SAS [Config-Lev1] Remote Services Stopped Manual
SAS [Config-Lev1] SAS Environment Manager Stopped Manual
SAS [Config-Lev1] SAS Environment Manager Agent Stopped Manual
SAS [Config-Lev1] SASApp - OLAP Server Stopped Manual
SAS [Config-Lev1] SASMeta - Metadata Server Stopped Manual
SAS [Config-Lev1] SASServer12_1 - WebAppServer Stopped Manual
SAS [Config-Lev1] SASServer1_1 - WebAppServer Stopped Manual
SAS [Config-Lev1] SASServer2_1 - WebAppServer Stopped Manual
SAS [Config-Lev1] Web Infrastructure Platform Data Server Stopped Manual
SAS [Config-Lev1] DIP JobRunner Stopped Manual
SAS [Config-Lev1] httpd - WebServer Stopped Manual
NOTE: SAS [Config-Lev1] SASMeta - Metadata Server is not in a Running state. Starting it.
NOTE: Start command returned successfully for SAS [Config-Lev1] SASMeta - Metadata Server.
NOTE: Checking status of SAS [Config-Lev1] SASMeta - Metadata Server to confirm it has started before moving on.
Current status: Start Pending
Current status: Start Pending
Current status: Start Pending
Current status: Start Pending
Current status: Running