

Using the SAS® Object Spawner Log to Identify Issues More Quickly and Avoid Downtime

Jessica Franklin, SAS Institute Inc.

ABSTRACT

Have you ever wondered why the log file is one of the first items that SAS Technical Support requests? The reason is that the log file contains a wealth of information about what actually occurs in the application or platform. So, what exactly is Technical Support looking for? How can the SAS® Object Spawner log help pinpoint the underlying issue? By reading this paper, SAS administrators will gain insight into how to use the object-spawner log to identify issues more quickly and avoid downtime. The focus of this paper is on the object-spawner log, because the object spawner is the gateway to other application servers. The paper provides an overview of the object-spawner log, describes useful loggers that you should be aware of, details how to identify common issues by looking in the logs, and offers solutions for those issues. With this knowledge, you can be proactive in solving problems that arise.

INTRODUCTION

When a problem occurs for a user, it is vital that it is fixed quickly so that they can continue working. After you determine that the issue is not with a metadata server, such as failed authentication, the next place to investigate is the servers on which the jobs are being run. The workspace server, pooled workspace server, and stored process server are all launched by the object spawner. So, the object-spawner log is the first place to start looking for any server issues.

This paper provides you with a starting point for becoming more aware of the benefits of understanding the object spawner and its logs. It helps new or experienced SAS administrators learn how to interpret the object-spawner logs so that they can get their user's jobs running again in a timely fashion.

This paper covers the following topics:

- the object-spawner log and the wealth of knowledge that it contains to help you understand and fix issues as they arise
- various loggers and how to use them
- common issues that are identified in the object-spawner logs and how to quickly fix them
- information to collect before contacting Technical Support

UNDERSTANDING THE OBJECT-SPAWNER LOG

This section discusses object spawners, console logs, rolling and initialization logs, and object-spawner connections.

OBJECT SPAWNER OVERVIEW

The *SAS® 9.4 Intelligence Platform: Application Server Administration Guide* provides the following definition for object spawner: “An object spawner runs on each machine where you want to run a workspace server or stored process server, listens for requests, and launches the servers as necessary.” The object spawner is the hub for all job requests to the workspace server, pooled workspace server, and stored process server. The object spawner is dependent only upon the metadata server and receives all the connection information for the metadata server from the metadataconfig.xml file.

THE OBJECT-SPAWNER CONSOLE LOG VERSUS THE OBJECT-SPAWNER LOG

The object-spawner console log, which can be seen on UNIX platforms, is not actually a log that contains information about the object spawner. Instead, this log contains information about the applications that are launched by the object spawner. When diagnosing object-spawner related issues, you should look at the object-spawner log, not the object-spawner console log. This paper does not discuss the object-spawner console log.

THE OBJECT-SPAWNER LOG

Here are the default locations for the object-spawner log:

- Windows default location: `SAS\Config\Lev1\ObjectSpawner\Logs`
- UNIX default location: `sas/config/Lev1/ObjectSpawner/Logs`

You can determine whether you have a customized location setup for where the object-spawner logs reside by looking at the logconfig.xml file located in the `SAS\Config\Lev1\ObjectSpawner\logconfig.xml` directory (UNIX: `sas/config/Lev1/ObjectSpawner/logconfig.xml`).

Look at the FileNamePattern to determine the location to which the object-spawner logs are being written. Here is an example from Windows:

```
<param name="FileNamePattern"
value="D:\SAS\Config\Lev1\ObjectSpawner\Logs\ObjectSpawner_%d_%S{hostname}_
%S{pid}.log"/>
```

Looking at the above example, you see the FileNamePattern parameter. This parameter shows that the log is created in the `D:\SAS\Config\Lev1\ObjectSpawner\Logs\` location and that the actual log name is `ObjectSpawner_%d_%S{hostname}_%S{pid}.log`. In the log name, %d refers to the date, {hostname} refers to the host name where the object spawner is running, and {pid} refers to the process ID (PID) that the object spawner is running under on the operating system.

THE OBJECT-SPAWNER LOG AT INITIALIZATION VERSUS THE ROLLING OBJECT-SPAWNER LOG

When you troubleshoot object-spawner issues, it is important to know which log you should look at—the object-spawner initialization log or the object-spawner rolling log. For example, if the issue occurs for a few users connecting to the workspace server, then you should look

at the current rolling log to see what is being reported when they are attempting to launch a workspace-server session. If the issue is that the object spawner fails to start or you do not see connections being made to a workspace server in the rolling logs, then you should look at the initialization log to see whether the server definitions had issues being read. Both logs are useful for troubleshooting.

When an object spawner is first started, the log name contains a unique PID number. The object spawner uses a rolling logging system in which a new object-spawner log is created every day. The same PID number is associated with each of these logs until the object spawner is restarted. After an object spawner is restarted, the log name contains a new PID number. So, you can identify an object-spawner initialization log versus an object-spawner rolling log by the PID number or by looking at the log itself.

Note that the header information, discussed below, is always the first line in any object-spawner log, either rolling or initialization.

The Object-Spawner Rolling Log

As previously stated, one way to determine whether the object-spawner log is a rolling log is to look at the PID number. If the PID number is the same as the previously dated object-spawner logs, then it is a rolling log. The other way to determine whether it is a rolling log is to open the log file and review its contents.

For example, Output 1 shows the contents of an object-spawner rolling log:

```
Host: 'SASHOST', OS: 'WIN', Release: '10.0', Command: '"C:\Program Files\SASHome\SASFoundation\9.4\objspawn"
Log continued from D:\SAS\Config\Levl\ObjectSpawner\Logs\ObjectSpawner_YYYY-MM-DD_14516.log
2018-11-09T00:00:00,120 INFO [00041164] :SYSTEM@SASHOST - New out call client connection (3108) for user sas
2018-11-09T00:00:00,124 INFO [00041164] :SYSTEM@SASHOST - Client connection 3108 for user sasevs@saspw close
2018-11-09T00:00:00,125 INFO [00041164] :sasevs@saspw - New client connection (3107) accepted from server po
2018-11-09T00:00:00,140 INFO [00041187] 3107:SYSTEM@SASHOST - Client connection 3107 for user sasevs@saspw c
2018-11-09T00:00:00,241 INFO [00041189] :SYSTEM@SASHOST - New out call client connection (3110) for user sas
2018-11-09T00:00:00,243 INFO [00041189] :SYSTEM@SASHOST - Client connection 3110 for user sasevs@saspw close
2018-11-09T00:00:00,243 INFO [00041189] :sasevs@saspw - New client connection (3109) accepted from server po
2018-11-09T00:00:00,255 INFO [00041212] 3109:SYSTEM@SASHOST - Client connection 3109 for user sasevs@saspw c
2018-11-09T00:00:00,353 INFO [00041214] :SYSTEM@SASHOST - New out call client connection (3112) for user sas
2018-11-09T00:00:00,355 INFO [00041214] :SYSTEM@SASHOST - Client connection 3112 for user sasevs@saspw close
2018-11-09T00:00:00,355 INFO [00041214] :sasevs@saspw - New client connection (3111) accepted from server po
2018-11-09T00:00:00,367 INFO [00041237] 3111:SYSTEM@SASHOST - Client connection 3111 for user sasevs@saspw c
2018-11-09T00:00:00,462 INFO [00041239] :SYSTEM@SASHOST - New out call client connection (3114) for user sas
```

Output 1. Example Object-Spawner Rolling Log

You can determine that the log shown in Output 1 is a rolling object-spawner log by looking at the second line printed in the log, which states "Log continued from ...". In this example, the log shows that it is a continuance of the log from the day before named ObjectSpawner_YYYY-MM-DD_14516.log. This line indicates that this log file is indeed a rolling object-spawner log.

The Object-Spawner Initialization Log

As previously stated, you can identify an object-spawner initialization log by looking at the PID number. If the PID number is different from the previous day's PID number, then it is an initialization log. You can also open the object-spawner log file to see whether it is an initialization log.

An example of an object-spawner initialization log is shown in Output 2:

```
0', Command: '"C:\Program Files\SASHome\SASFoundation\9.4\objspawn" -name "SAS [Config-Lvl] Object Spawner"'
:SYSTEM@SASHOST - SAS Object Spawner Daemon III version 9.40.20170906 is initializing.
:SYSTEM@SASHOST - Command line (0x0) C:\Program Files\SASHome\SASFoundation\9.4\objspawn
:SYSTEM@SASHOST - Option (0xc0000007) name
:SYSTEM@SASHOST - value (0x0) SAS [Config-Lvl] Object Spawner
:SYSTEM@SASHOST - Option (0x4) dnsMatch
:SYSTEM@SASHOST - value (0x0) SASHOST.na.sas.com
:SYSTEM@SASHOST - Option (0xe) sasspawnercn
:SYSTEM@SASHOST - value (0x0) Object Spawner - SASHOST
:SYSTEM@SASHOST - Option (0x10) xmlconfigfile
:SYSTEM@SASHOST - value (0x0) D:\SAS\Config\Levl\ObjectSpawner\metadataConfig.xml
:SYSTEM@SASHOST - Option (0x80000020) sspi
:SYSTEM@SASHOST - Option (0x8000000f) logconfigloc
:SYSTEM@SASHOST - value (0x0) D:\SAS\Config\Levl\ObjectSpawner\logconfig.xml
:SYSTEM@SASHOST - Processing configuration file (D:\SAS\Config\Levl\ObjectSpawner\metadataConfig.xml).
:SYSTEM@SASHOST - New out call client connection (1) for user sastrust@saspw. Encryption level is Everything u
:SYSTEM@SASHOST - Objspawn is executing on host SASHOST.SAS.com (ab80::a489:5c8:cd2d:2efg%4).
:SYSTEM@SASHOST - Also known as:
:SYSTEM@SASHOST - localhost
:SYSTEM@SASHOST - SASHOST
:SYSTEM@SASHOST - view-localhost
:SYSTEM@SASHOST - ab80::a489:5c8:cd2d:2efg%4
:SYSTEM@SASHOST - ab80::a489:5c8:cd2d:2efg%6
:SYSTEM@SASHOST - ab80::1c56:b841:9567:d345%11
:SYSTEM@SASHOST - 10.11.12.13
:SYSTEM@SASHOST - 111.122.123.1
:SYSTEM@SASHOST - 111.122.31.1
:SYSTEM@SASHOST - ::1
:SYSTEM@SASHOST - 127.0.0.1
:SYSTEM@SASHOST - Processing parsed configuration results.
:SYSTEM@SASHOST - Objspawn is proceeding with the configuration results processing utilizing the A5WN99NR.AZ000
```

Output 2. Example Object-Spawner Initialization Log

Here are the key points to identify that Output 2 shows an initialization log:

- The second line in the log does not state "Log continued from ..."
- The second line in the log states "SAS Object Spawner Daemon III version 9.40.xxxxxx is initializing."

The Object-Spawner Initialization Log

The object-spawner initialization log contains a lot of information that you cannot find in the rolling logs. The first line, referred to as the header information, contains launch information for the object spawner. This header information is always printed at the top of every object-spawner log, either rolling or initializing. The lines that follow this header give more details as the object spawner initializes.

Object-Spawner Header Information

Header information is always present in object-spawner logs and is the first line written. The information detailed below is presented in this order in the log.

Host Name: The first item in the header information is the host name that the object spawner is running on. Here is an example: Host: 'SASHOST'. The host name is returned by the host itself.

OS and Release: Next, you see the type of operating system that the object spawner is running on and which release of the operating system is running. Here is an example: OS: 'WIN', Release: '10.0'.

Launch Command: The launch command points to the objspawn executable and is used to launch the object spawner.

Additional Options Set: The next set of items shows the CMD_OPTIONS that are set in the ObjectSpawner.bat file or the ObjectSpawner.sh file. These options can include DNSMATCH, SASSPAWNERCN, XMLCONFIGFILE, SSPI, LOGCONFIGLOC, and so on.

The Object-Spawner Initialization Process

The initialization process starts with the second line that states "SAS Object Spawner Dameon III version 9.40.xxxxxx is initializing." Next, it lists the values for the current options that are set.

Reading the Executable File

The object-spawner executable file is used to start the object spawner. The executable file for the object spawner can be found in the following default locations:

Windows: C:\Program Files\SASHome\SASFoundation\9.4\objspawn.exe

UNIX: /sas/sashome/SASFoundation/9.4/utilities/bin/objspawn

NAME: (Windows only) Refers to the service name that is defined in the ObjectSpawner.bat file.

DNSMATCH: Refers to the CMD_OPTION that is set in the ObjectSpawner.bat or ObjectSpawner.sh file. This option specifies the local machine that the object spawner accepts as the Domain Name System (DNS) alias. The DNSMATCH value is replaced by the local machine name in the spawner's list of servers.

HOSTKNOWNBY: Refers to the CMD_OPTION that is set in the ObjectSpawner.bat or ObjectSpawner.sh file. This option specifies the local machine that the object spawner accepts as the DNS alias.

Note: You might declare either the DNSMATCH or the HOSTKNOWNBY option, but you cannot declare both. The DNSMATCH and HOSTKNOWNBY options function the same by allowing the object spawner to accept the DNS alias as a match for the local machine. However, you must use DNSMATCH if you have multiple nodes that run object spawners and your network configuration resolves only a single DNS alias.

SASSPAWNERCN: Refers to the name of the object-spawner object in the metadata. This name is set in the ObjectSpawner.bat or ObjectSpawner.sh file. If SASSPAWNERCN is not specified, the object spawner uses the first spawner definition that matches the current host on the metadata server.

XMLCONFIGFILE: Refers to the location of the metadataconfig.xml file where the object spawner picks up connection information for the metadata server.

SSPI: Refers to an option that is set if you use single sign-on connections to the object spawner.

LOGCONFIGLOC: Refers to the location for the logconfig configuration file that contains information about where to write the object-spawner logs and the loggers and logging levels that it should be using.

For more spawner invocation options that are available, see the "[Spawner Invocation Options](#)" section in *SAS® 9.4 Intelligence Platform: Application Server Administration Guide*.

Processing the Metadataconfig.xml File

One of the first actions that the object spawner takes, after processing the executable file, is to connect to the metadata server. It does this by reading its metadataConfig.xml file that contains information about how to connect to the metadata server. Output 3 shows this process:

```
Processing configuration file
(C:\SAS\Config\Lev1\ObjectSpawner\metadataConfig.xml). ❶
New out call client connection (1) for user sastrust@saspw. Encryption
level is Everything using encryption algorithm AES. Peer IP address and
port are [ab80::a489:5c8:cd2e:2fgh%4]:8561. ❷
```

Output 3. Excerpt from Object-Spawner Initialization Log File

- ❶ The object spawner is reading information about how to connect to the metadata server.
- ❷ This line shows when the object spawner is making its initial connection to the metadata server.

Understanding Host-Name Information, Ports, and Server Connections

After the object spawner has successfully connected to the metadata server, it proceeds with initialization by defining the host that it is running on, reserving server ports, reading server definitions from metadata, and actively listening on the ports.

```
:sas - Objspawn is executing on host SASHOST.unx.sas.com (10.11.12.13).
:sas - Also known as:
:sas -     localhost
:sas -     SASHOST.sas.com
:sas -     10.11.12.13
:sas -     127.0.0.1
:sas - Processing parsed configuration results.
:sas - Objspawn is proceeding with the configuration results processing utilizing the A50BLOWY.AZ000003
:sas - Reserved IPv6 port 8581 for administrator listen (connection 2).
:sas - Reserved IPv6 port 8451 for server listen (connection 3).
:sas - Reserved IPv6 port 8701 for server listen (connection 4).
:sas - Reserved IPv6 port 8601 for server listen (connection 5).
:sas - Reserved IPv6 port 8611 for server listen (connection 6).
:sas - Reserved IPv6 port 8621 for server listen (connection 7).
:sas - Reserved IPv6 port 8631 for server listen (connection 8).
:sas - Reserved IPv6 port 8591 for server listen (connection 9).
:sas - Reserved IPv6 port 33193 for launched server connect back listen (connection 10).
:sas - Activated listen on IPv6 port 33193 (connection 10).
:sas - Cluster SASApp - Logical Pooled Workspace Server (A50BLOWY.AX000005) containing servers of type 6
:sas - Cluster SASApp - Logical Stored Process Server (A50BLOWY.AX000006) containing servers of type 159
:sas - Activated listen on IPv6 port 8581 (connection 2).
:sas - Cluster SASApp - Logical Stored Process Server (A50BLOWY.AX000006) has property PerClientCost=1.
```

Output 4. Object-Spawner Initialization Tasks

Host-Name Information: Host-name information about where the object spawner executes is next in the log file. This line shows the actual host name that the object-spawner is running on, including all the other names that host is known by.

Reserved Listening Ports: The messages "Reserved IPvX port XXXX for server listen" indicates that the object spawner binds to a port.

Pooled Workspace Server and Stored Process Server: If you have any pooled workspace servers or stored process servers assigned to this particular object spawner, this line is where you see the definitions being picked up. It also includes the information about which load-balancing algorithm the servers are using.

If you have set up load balancing on a workspace server, you will see it listed in this area of the log file. It also shows which load-balancing algorithm it is using.

Activated Listen on Ports: The message "Activated listen on IPvX port XXXX" indicates that the object spawner is actively listening on those ports now.

Object Spawner Initialization Complete: Initialization completes when you see the messages shown in Output 5:

```
SAS Object Spawner Daemon III is running under the user identity
SYSTEM@SASHOST.
Object Spawner Daemon III has completed initialization.
SAH021999I Server SAS Object Spawner Daemon III (8581), State, running
```

Output 5. Log Messages That Show That Initialization Is Complete

CONNECTIONS MADE TO THE OBJECT SPAWNER

After the object spawner has completed its initialization, the log shows new connections coming in. The following example shows object-spawner log output from running `proc setinit; run;` from SAS® Enterprise Guide®. After Output 6, you see a line-by-line analysis of what occurs in the log.

```
sasdemo@SASHOST - New client connection (44729) accepted from server port
8591 for SAS user sasdemo@SASHOST.❶ Encryption level is Everything using
encryption algorithm AES.❷ Peer IP address and port are
[abcd::f123:5a2:ee2c:2eac%4]:50157 for APPNAME=SAS Enterprise Guide.❸
sasdemo@SASHOST - Created process 31900 using credentials sasdemo@SASHOST
for user sasdemo@SASHOST (child id 5).❹
SYSTEM@SASHOST - New out call client connection (39941) for launched server
(child 5). Peer IP address and port are
[fe80::f489:5b8:ec2c:2eac%4]:62087.❺
SYSTEM@SASHOST - Launched process 31900 (child id 5) is now running as
process 20932.❻
SYSTEM@SASHOST - Client connection 44729 for user sasdemo@SASHOST closed.❼
```

Output 6. Object-Spawner Log Output

- ❶ The object spawner accepts the connection from the client for the sasdemo user since sasdemo was the user ID that logged into SAS Enterprise Guide and requested a workspace-server session. You can tell that the connection is to a workspace server based on the port number. Out of the box, the workspace-server port number is set to 8591.
- ❷ The connection uses the AES encryption algorithm with an encryption level set to **Everything**.
- ❸ SAS Enterprise Guide is the client that requested a workspace-server session.
- ❹ The object spawner creates the workspace-server process for sasdemo.
- ❺ A connection between the launched workspace-server process and the object spawner is established.
- ❻ (Windows only) The first PID, 31900, is for the execution of the WorkspaceServer.bat file. PID 20932 is for the actual workspace-server process.
- ❼ The client inherits the connection to the workspace server and closes its connection with the object spawner.

OBJECT-SPAWNER REFRESH

Unlike a restart, when you perform an object-spawner refresh, it does not kill any currently running jobs. This section describes what occurs in the object-spawner logs when you perform a refresh.

For instructions about how to refresh an object spawner, see “[Refresh the Object Spawner](#)” in the *SAS® 9.4 Intelligence Platform: Application Server Administration Guide*.

When you refresh an object spawner in SAS® Management Console, you see messages similar to what is shown in Output 7 in the logs:

```
sasadm@saspw - START
Perf.ARM.IOM.ObjectSpawner.ServerAdministration.ListClients 4fldcc0 0 0
sasadm@saspw - STOP
Perf.ARM.IOM.ObjectSpawner.ServerAdministration.ListClients 0 0
sasadm@saspw - SAH023999I Server SAS Object Spawner Daemon III (8581),
State, paused ❶
sasadm@saspw - SAH021999I Server SAS Object Spawner Daemon III (8581),
State, running ❷
sasadm@saspw - New out call client connection (45) for user sastrust@saspw.
Encryption level is Everything using encryption algorithm AES. Peer IP
address and port are [fe80::f489:5b8:ec2c:2eac%4]:8561. ❸
sasadm@saspw - Processing parsed configuration results.
sasadm@saspw - Objspawn is proceeding with the configuration results
processing utilizing the A5WN99NR.AZ000003 spawner definition.
sasadm@saspw - Reserved IPv6 port 8581 for administrator listen (connection
46).
SYSTEM@SASHOST - Cluster SASApp - Logical Pooled Workspace Server
(A5WN99NR.AX000005) containing servers of type 620963ee-32bf-4128-bf5f-
4b0df8ff90eb using algorithm Most Recently Used found.
SYSTEM@SASHOST - Cluster SASApp - Logical Stored Process Server
(A5WN99NR.AX000006) containing servers of type 15931E31-667F-11D5-8804-
00C04F35AC8C using algorithm Cost found.
SYSTEM@SASHOST - Activated listen on IPv6 port 8581 (connection 46).
SYSTEM@SASHOST - Cluster SASApp - Logical Stored Process Server
(A5WN99NR.AX000006) has property PerClientCost=1.
SYSTEM@l110d195 - Processor initialized. ❹
```

Output 7. Object-Spawner Log Output after an Object-Spawner Refresh

- ❶ The object spawner goes in a paused state and queues any requests that are received. It does not kill the object-spawner process or any server processes that are currently running.
- ❷ The object spawner is running again and is not in a paused state.
- ❸ The object spawner connects to the metadata server again.
- ❹ The object spawner rereads the metadata. When the object spawner rereads the metadata, it reserves and actively listens on the ports again and reads the server definitions. After that, the object spawner is operational again and ready for connections. Any requests that were queued are handled at this time.

OBJECT-SPAWNER STOP

When an object spawner is intentionally stopped, the logs resemble Output 8 below:

```
SYSTEM@SASHOST - START
Perf.ARM.IOM.ObjectSpawner.ServerAdministration.StopServer 425f6a0 0 0 ❶
SYSTEM@SASHOST - Processor has been stopped.
SYSTEM@SASHOST - Orphaning server SASApp - Workspace Server in process
7940. ❷
SYSTEM@SASHOST - The server is not running (paused/deferred stop mode?). ❸
SYSTEM@SASHOST - Client connection 30163 for user sassrv@SASHOST closed.
SYSTEM@SASHOST - Client connection 22772 for user sastrust@saspw closed. ❹
SYSTEM@SASHOST - Processor has been destroyed.
SYSTEM@SASHOST - STOP
Perf.ARM.IOM.ObjectSpawner.ServerAdministration.StopServer 0 0
SYSTEM@SASHOST - Bridge PE SAM listen thread is exiting (0).
SYSTEM@SASHOST - For the IOM thread puddle, there were 534367 requests
processed. The maximum number of requests queued was 0. The maximum number
of worker threads was 5.
SYSTEM@SASHOST - Bridge protocol engine has quiesced.
SYSTEM@SASHOST - Bridge protocol engine is unloading.
SYSTEM@SASHOST - SAH029999I Server SAS Object Spawner Daemon III (8581),
State, stopped ❺
```

Output 8. Object-Spawner Log Output When the Object Spawner Is Intentionally Stopped

- ❶ This is the first instance of the stop of the process being called.
- ❷ The processor is stopped, which prevents new workspace-server requests from being handled. The object spawner tells all launched processes to stop. You might see some processes being orphaned. In Output 8, SAS Enterprise Guide is not closed. Therefore, the workspace-server process is still running. Because the process is still running, the object spawner severs ties with the server process and orphans it. You see the orphaned message for any currently running server processes.
- ❸ The object spawner is stopped, but it is waiting on existing client connections to close.
- ❹ Any client connections that are still open will be closed. Depending on how many connections you have, you might see quite a few of these client-connection-closed messages.
- ❺ These are the final messages before the object spawner is officially stopped in the last line of the log.

TURNING UP THE LOGGERS

This section describes object-spawner loggers. These loggers can help when you need to produce more information about what is happening in the background.

STATIC LOGGING IN THE OBJECT-SPAWNER CONFIG FILES VERSUS DYNAMIC LOGGING IN SAS® MANAGEMENT CONSOLE

There are two different ways in which you can change loggers and their logging levels in SAS: Static logging or dynamic logging.

Static Logging

Static object-spawner logging refers to when you make changes to the logconfig.xml files in the object-spawner configuration directory. Static logging is useful if you need to capture additional information on an intermittent problem. You must restart the object spawner to enable the static logging. To disable static logging, you must remove it from the file and restart the object spawner. Note that static logging is not disabled until you perform both of these actions.

Dynamic Logging

Dynamic object-spawner logging is available in SAS® 9.2 and later. You can enable dynamic logging in SAS Management Console. Dynamic logging is helpful if you are unable to restart your object spawner due to ongoing business operations or if the problem goes away when the object spawner is restarted. Dynamic logging remains enabled until you disable it or until you restart the object spawner.

TRACE-LEVEL LOGGING WITH THE OBJECT-SPAWNER APP LOGGER

Trace-level logging is helpful when you are investigating an issue. When trace-level logging is enabled, it produces more detailed information. There are many loggers that you can enable, depending on what you are trying to accomplish.

Trace-Level Logging on the App Logger

The main logger that you use trace-level logging on is the App logger. This logger produces more details about everything that occurs in the object spawner. The App logger is typically the first logger that Technical Support asks you to enable trace-level logging on.

For example: You see an error message when starting the object spawner that states "port already in use." By turning the App logger to **Trace**, it shows more details about which process is using which ports. You can then search for the port number to see whether, in fact, the port is already being used. When you are finished diagnosing the issue, you can turn off trace-level logging by reversing the steps that you performed.

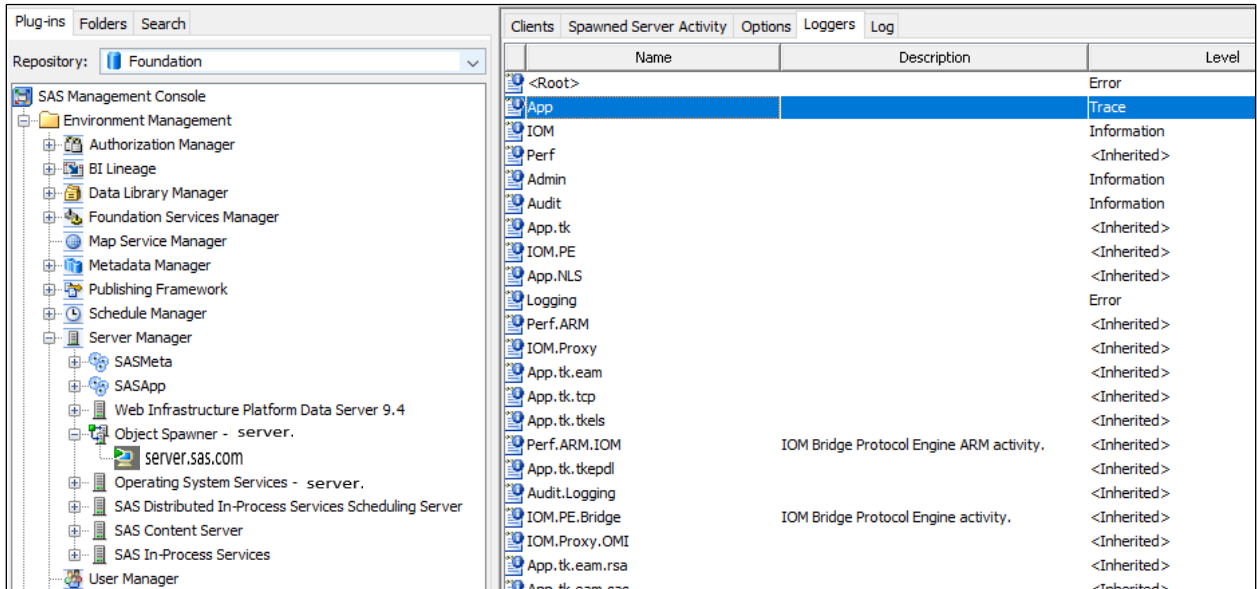
Procedure to Enable Trace-Level Logging

To enable **static trace-level logging** on the object spawner for the App logger, navigate to your equivalent of `c:\SAS\Config\Lev1\ObjectSpawner\Logconfig.xml`. Then, follow the steps in the "[Enable More Detailed Logging for SAS Object Spawner Troubleshooting](#)" section of *SAS® 9.4 Intelligence Platform: System Administration Guide, Fourth Edition*.

To enable **dynamic trace-level logging** for the App logger in SAS Management Console, complete these steps:

1. Open SAS Management Console.
2. Expand **Server Manager ► Object Spawner – XXXX**.
3. Right-click the node name under **Object Spawner – XXXX**.
4. Select **Connect**.
5. In the right pane, click the **Loggers** tab.
6. Locate and right-click **App**, and then select **Properties**.
7. Change the **Assigned** selection to **Trace**.

- Click **OK**. You should see **Trace** in the **Level** column of the **Loggers** tab as shown in Display 1 below.



Display 1. The Object Spawner Loggers Tab in SAS Management Console

- When you complete your troubleshooting, you must follow these steps again but select **Information** for the **Level** radio button.

OTHER LOGGERS MOST COMMONLY ENABLED BY TECHNICAL SUPPORT

You can use the loggers described in this section to look more in depth at what is occurring with the application servers. Most often, after you enable these loggers, you still need to [contact SAS Technical Support](#) to read the output because it is difficult to understand and interpret. If you are experiencing any issues surrounding these areas, it helps to turn the loggers to **Trace**, reproduce the issue, and then attach the object-spawner log to your Technical Support tracking entry. Following this process reduces the time that SAS Technical Support spends starting the investigation.

Audit.Authentication Logger

You can use the Audit.Authentication logger to isolate authentication issues. When this logger is enabled, you can see more details related to the server-authentication requests and find where in the process the issue is occurring.

App.tk.tkegrid

You can use the App.tk.tkegrid logger when your environment is grid-launched to debug a potential grid-launched server issue. When the workspace server, stored process server, or pooled workspace server are grid enabled, a unique grid file is not created because these servers do not use the sasgrid script. By enabling trace-level logging on the App.tk.tkegrid logger, you can produce more information in the object spawner to show what is occurring when these grid-enabled servers are starting.

IOM.LoadBalancing

You can enable the IOM.LoadBalancing logger when you have load balancing enabled in your system. It produces more details about the session count, based on the load-balancing algorithm that you have selected. This logger is helpful to determine whether you are running out of job slots and need to adjust your load-balancing algorithm.

To learn more about loggers, see the "[Loggers](#)" section in *SAS® 9.4 Logging: Configuration and Programming Reference, Second Edition*.

IDENTIFYING COMMON ISSUES AND THEIR SOLUTIONS

This section covers how you use the log to identify common issues. It describes error messages that you could see in the log, what the error messages can indicate, possible solutions, and links to SAS Notes (if available) that provide step-by-step instructions for how to address the issue.

OBJECT SPAWNER FAILS TO START

If the object spawner fails to start, the first place to investigate is the most recent object-spawner log after you attempted to start it. If no object-spawner log is being produced, it might indicate that the object-spawner launch credentials do not have Write permissions to the object-spawner logging location.

Access Denied

The object spawner fails to start, and you see an "Access denied" error in latest object-spawner log from after you attempted to start the object spawner.

Error Messages

```
2019-1-25T06:02:49,203 INFO [00000004] :sas - Processing configuration file
(/sas/config/Lev1/ObjectSpawner/metadataConfig.xml) .
2019-1-25T06:02:49,232 ERROR [00000008] :sas - Access denied.
2019-1-25T06:02:49,233 ERROR [00000008] :sas - Unable to connect to the
metadata server.
```

Scenario 1

One reason that this error occurs is that the sastrust password is incorrect in the `/sas/config/Lev1/ObjectSpawner/metadataConfig.xml` file. To circumvent this issue, run SAS® Deployment Manager to update the password. This action updates the password in the metadata as well as in any configuration files where the password might be incorrect.

Scenario 2

Another possible cause for the object spawner failing to start is corrupted metadata. You can run the [Metadata Analyze/Repair Tools](#) if needed to address this issue. If the object spawner still fails to start after you run this action, [contact SAS Technical Support](#).

Host-Name Mismatch

The object spawner fails to start, and you see the following error in the latest object-spawner log after you attempted to start the object spawner.

Error Messages

The A56YVPDQ.AY000002 spawner definition is not defined for this machine and will be ignored.

The Object Spawner - XXXX spawner definition did not exist in the processed configuration. Objspawn cannot continue without a spawner definition.

CLASSFACTORY cannot be instantiated.

Scenario

If the host name defined in the metadata does not match the value specified for DNSMATCH or any of the host names returned from the operating system, then there is a host-name mismatch. Look at an object-spawner initialization message for the section in Output 9 below to determine whether the names do not match.

```
Objspawn is executing on host SASHOST.na.SAS.com (ab80::c123:4b8:ab2c:2def%4).
Also known as:
  localhost
  SASHOST
  view-localhost
  ab80::c123:4b8:ab2c:2def%4
  11.22.33.444
  111.111.31
  ::1
  127.0.0.1
```

Output 9. Operating System Returned Host Names

To circumvent the issue, complete the steps below for your environment.

In **Windows environments**, complete the following steps:

1. Stop the Object Spawner services: `ObjectSpawner.bat stop`
2. Remove the Object Spawner service: `ObjectSpawner.bat remove`
3. In the ObjectSpawner.bat file, change the `Set CMD_OPTIONS= -dnsmatch server.sas.com` line to the following: `Set CMD_OPTIONS= -dnsmatch server`
4. Re-install the object-spawner service: `ObjectSpawner.bat install`
5. In SAS Management Console, navigate to **Server Manager ► Object Spawner - XXXX ► Properties ► Options** tab, create a new machine named "server," and move it under the **Selected** list.
6. Repeat this procedure for SASApp - Workspace Server, SASApp - Stored Process Server, SASApp - Pooled Workspace Server, and SASApp - Visual Process Orchestration Design Server. (You might have more server definitions to change.)
7. In SAS Management Console, navigate to **Server Manager ► Object Spawner - XXXX ► Properties ► Servers** tab and make sure that workspace server, pooled workspace server, and so on are included under the **Selected** list.
8. Start the object-spawner service.
9. Validate the workspace server, pooled workspace server, and stored process server.

In **Linux environments**, complete the following steps:

1. Navigate to your equivalent of the `/sas/config/Lev1/ObjectSpawner/ObjectSpawner.sh` file and change the `CMD_OPTIONS= -dnsmatch server.sas.com` line to the following: `CMD_OPTIONS= -dnsmatch server`
2. In SAS Management Console, navigate to **Server Manager ► Object Spawner - XXXX ► Properties ► Options** tab, create a new machine named "server," and move it under the **Selected** list.
3. Repeat this procedure for SASApp - Workspace Server, SASApp - Stored Process Server, SASApp - Pooled Workspace Server, and SASApp - Visual Process Orchestration Design Server. (You might have more server definitions to change.)
4. In SAS Management Console, navigate to **Server Manager ► Object Spawner - XXXX ► Properties ► Servers** tab and make sure that workspace server, pooled workspace server, and so on are included under the **Selected** list.
5. Start the object-spawner service.
6. Validate the workspace server, pooled workspace server, and stored process server.

Server Component Not Found

The object spawner fails to start, and you see the error "Server component not found" in the latest object-spawner log, after you attempted to start it.

Error Messages

```
Server component not found,  
servercomponent="omsobj:servercomponent?@name='Object Spawner - XXXX'"  
Unable to initiate the IOM run-time.
```

Scenario 1

You typically see this error message if the object-spawner name is spelled incorrectly.

To circumvent this issue, complete these steps:

1. In SAS Management Console, check the name of the object spawner by expanding **Server Manager**, right-clicking **Object Spawner**, and selecting **Properties**.
2. Make sure that the name (including all necessary spaces) listed reads exactly like the following:

```
Object Spawner - XXXX
```

Scenario 2

Another reason that you encounter this error is if the sastrust user does not have Read Metadata permissions on the object spawner.

To circumvent this issue, complete the following steps:

1. In SAS Management Console, expand Server Manager.
2. Right-click **Object Spawner - XXXX**, select Properties, and click the **Authorization** tab.

3. Make sure that `Grant` is selected for **ReadMetadata** for both SAS General Servers and SAS System Services.
4. Start the object spawner.

Scenario 3

You might also see this error if the metadata is corrupted. In this case, you need to [contact SAS Technical Support](#) for further analysis and support.

WORKSPACE SERVER FAILS TO VALIDATE

This section discusses workspace-server validations. If the workspace server fails to validate, you must check the latest log to see why it fails.

No Entries for the Workspace Server in the Object-Spawner Log

After you validate the workspace server in SAS Management Console, the object-spawner log does not show that any `ConnectionService` attempt is made. You should see something similar to the following (notice that this message is not listed for the time that you performed the validation):

```
New client connection (123) accepted from server port 8591 for user
sasdemo@SASHOST. Encryption level is Credentials using encryption
algorithm SAS Proprietary. Peer IP address and port are
[::ffff:10.11.12.13]:52605 for APPNAME=ConnectionService 904500.
```

Error Messages

The object-spawner initialization log does not show any entries for a workspace-server connection attempt.

Scenario 1

The lack of entries indicates that the object spawner is not reading the workspace-server definition.

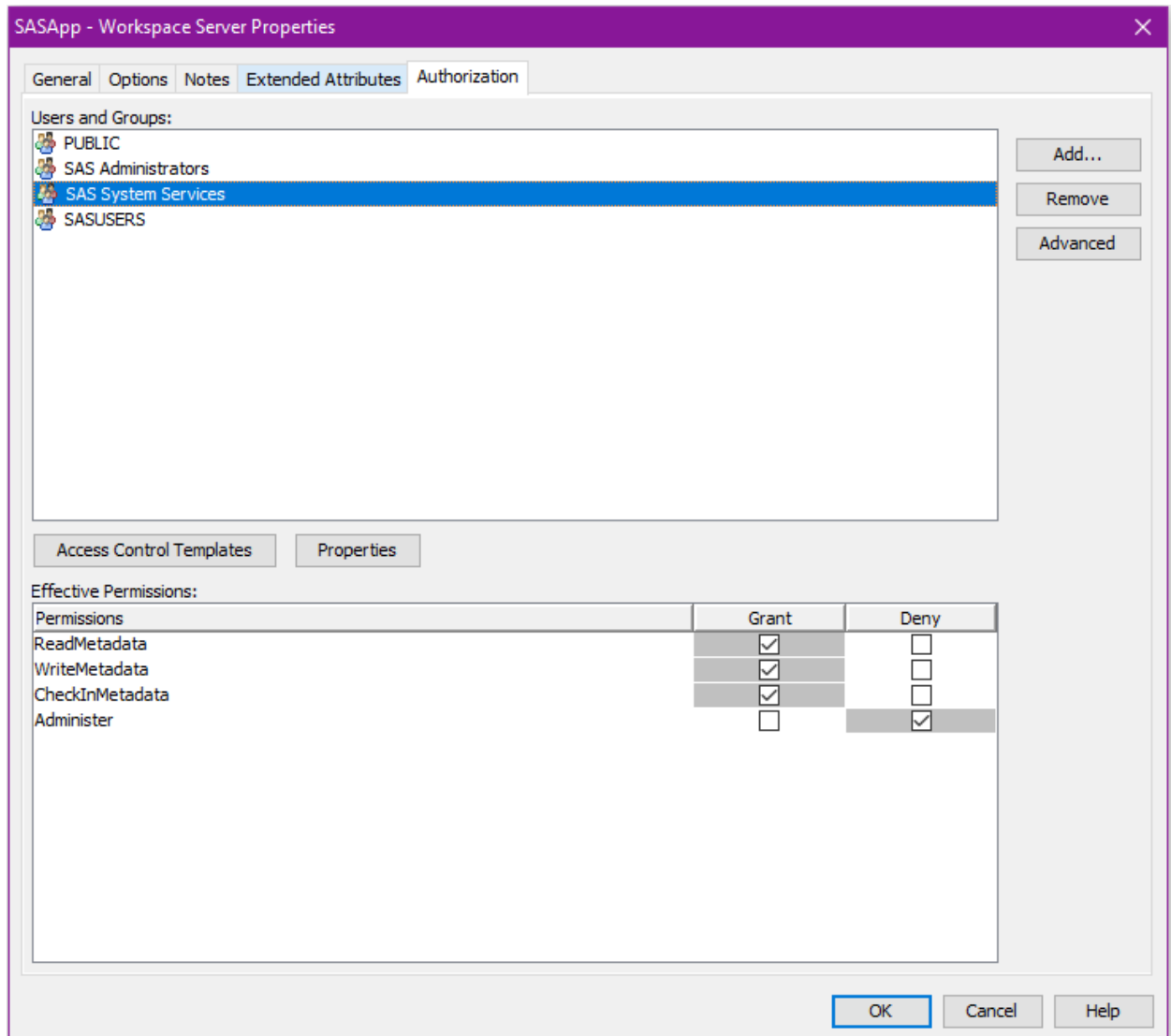
To circumvent this issue, you must add the workspace server to the **Selected** list in the object-spawner properties by completing these steps.

1. In SAS Management Console, expand Server Manager.
2. Right-click **Object Spawner – XXXX** and select **Properties**.
3. Select the **Servers** tab.
4. Make sure that the SASApp – Workspace Server is under the **Selected** servers list.
5. Click **OK**.
6. Restart or refresh the object spawner so that it picks up the server definition.

Scenario 2

A lack of entries can also indicate that the SAS Trusted User does not have Read Metadata authorization on the logical or lower level for the workspace server. To address this issue, complete the following steps:

1. In SAS Management Console, expand Server Manager.
2. Expand SASApp.
3. Expand SASApp – Logical Workspace Server, right-click **SASApp – Workspace Server**, and select Properties.
4. Click the **Authorizations** tab and make sure that SAS System Services has **Grant** selected for **ReadMetadata** as shown in Display 2 below.



Display 2. Permissions on the Authorizations Tab

5. Verify this permission at the SASApp – Logical Workspace Server level as well.

Scenario 3

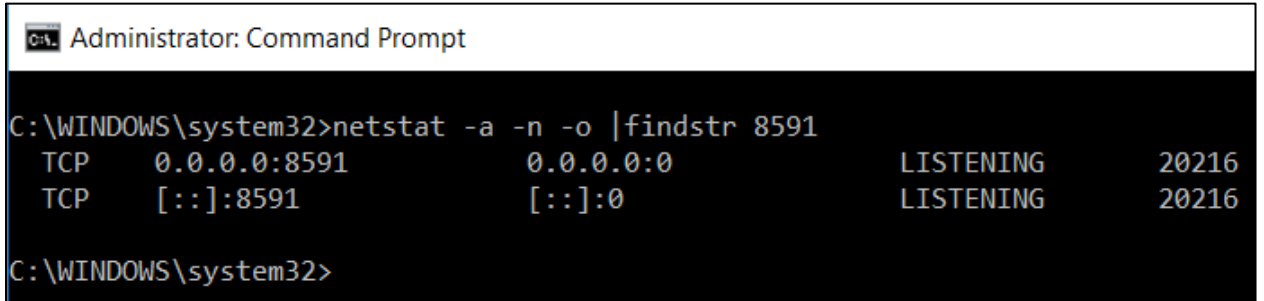
The client application is unable to get through to the object-spawner listen port. To circumvent this issue, complete the following steps:

1. Run the following commands to determine whether the object spawner is listening on the server port:

UNIX: `sudo netstat -nlp |grep 8591`

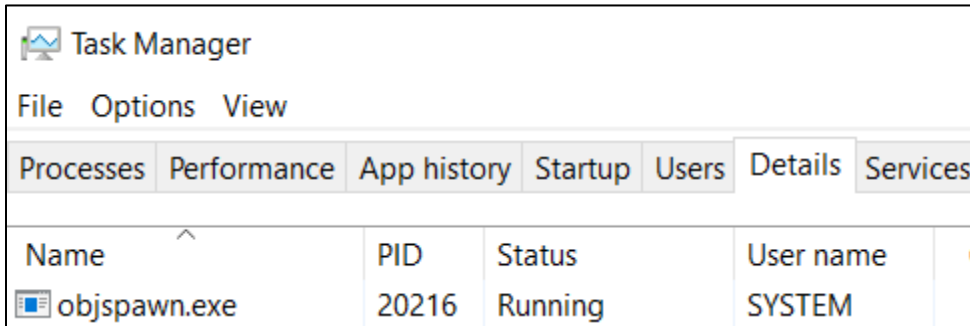
```
[sas@v64-01 WorkspaceServer]$ sudo netstat -nlp |grep 8591
tcp        0      0  :::8591                :::*                    LISTEN     14900/objspawn
[sas@v64-01 WorkspaceServer]$
```

Windows Command Line: `netstat -a -n -o |findstr 8591`



```
Administrator: Command Prompt
C:\WINDOWS\system32>netstat -a -n -o |findstr 8591
TCP        0.0.0.0:8591          0.0.0.0:0             LISTENING   20216
TCP        [::]:8591           [::]:0                LISTENING   20216
C:\WINDOWS\system32>
```

Windows Task Manager: Open Task Manager, click the **Details** tab, and look for a process named objspawn.exe.



2. If the object spawner is not listening on the server port, then review the object-spawner initialization logs for the object-spawner process. Look for any information that would explain why the server definition was not picked up. Some examples are that the port is in use or the credentials failed to authenticate (if token authentication is enabled). If there is no relevant information in the object-spawner initialization log, sastrust might not have permission to read the server definition or the server might not be in the object-spawner selected server list.
3. If the object-spawner is listening on the server port, then check whether the port is open on the firewall between the server and client.

Error 1385

When validating the workspace server, you see an "Error 1385 Access Denied" message in the object-spawner logs.

Error Messages

```
Error authenticating user userid in function LogonUser. Error 1385 (Logon failure: the user has not been granted the requested logon type at this computer. ). Access denied.
```

Scenario

Users do not have `Logon as a Batch Job` rights. This error message is specific to environments where SAS is running on a Windows server. To circumvent the issue, add the users or group to the `Logon as a Batch Job` rights in the operating system. Note, you need system administrative privileges to perform this action. If you do not have these privileges, get your system administrator to make the changes.

Userid@domain Does Not Have Permission

When validating the workspace server with a particular user ID, it fails to validate and you see that there is a permissions issue in the object-spawner log.

Error Messages

```
Client userid@domain does not have permission to use server SASApp -  
Workspace Server.
```

Scenario 1

This error occurs when the user does not have authorization to use the server. To circumvent the issue, check the authorizations in SAS Management Console for the SASApp – Workspace Server.

Scenario 2

Another reason that the error occurs is that the object spawner has a cached decision that needs to be cleared.

To circumvent the issue, use the **Clear Spawner Credentials Cache** action on the object spawner:

1. In SAS Management Console, expand Server Manager.
2. Expand Object Spawner – XXXX and right-click the node name.
3. Select **Connect**.
4. Right-click the node name again and select **Clear Spawner Credentials Cache**.

STORED PROCESS SERVER AND POOLED WORKSPACE SERVER FAIL TO VALIDATE

The stored process server and pooled workspace server are different because they are token servers. They use a launch credential called `sassrv` by default. A workspace server with token authentication enabled can also use the `sassrv` credentials.

Error Authenticating User SASSRV

The stored process server AND the pooled workspace server fail to validate. If they both fail to validate, in most cases, this indicates an issue with the `sassrv` credentials.

Error Messages

```
Error authenticating user domain\sassrv in function LogonUser. Error 1326  
(The user name or password is incorrect. ).
```

The credentials specified for the SASApp - Pooled Workspace Server (A5WN99NR.AZ000007) server definition failed to authenticate. Therefore this server definition will not be included.

Scenario

This error indicates that you might have an invalid sassrv password. To circumvent the issue, check the password in the operating system for sassrv by attempting to log on to the server directly as sassrv. If this action is successful, check the metadata and enter the password for sassrv so that it is the same.

1. In SAS Management Console, select **User Manager**.
2. Locate the SAS General Servers group and double-click it.
3. Click the **Accounts** tab.
4. Select the stored sassrv entry, and then select **Edit**.
5. Update the password to what was successful on the operating system.
6. Restart the object spawner.
7. Check the logs again to verify that the error is gone.

Unrestricted Access

The stored process server and pooled workspace server fail to validate, and you find an error indicating that the sassrv credentials have been given unrestricted access in the metadata.

Error Messages

```
Error authenticating user domain\sassrv in function LogonUser. Error 1326
(The user name or password is incorrect. ).
```

```
credentials specified for the SASApp - Stored Process Server
(A5WN99NR.AZ000008) server definition failed to authenticate. Therefore this
server definition will not be included.
```

```
The password obtained from the metadata indicates that it was obtained by a
user that has unrestricted access. Please check your configuration.
```

```
Unrestricted access can include being an unrestricted user, or being a member
of the SAS Administrators group, or being a member of any group that has the
role Metadata Server: User Administration.
```

Scenario

The error occurs because the sassrv or sastrust users are unrestricted. To circumvent the issue, make sure that sassrv and/or sastrust are not inheriting unrestricted access from a role or group in SAS Management Console. See [SAS Note 58183](#) for helpful tips about where to start investigating.

OTHER COMMON ISSUES

You can diagnose many more issues using messages in the object-spawner log. For more information about these messages, see the "[Object Spawner Messages](#)" section in SAS® 9.4 *Intelligence Platform: Application Server Administration Guide*. You can also search support.sas.com to find SAS Notes regarding error messages.

COLLECTING RELEVANT INFORMATION BEFORE CONTACTING SAS TECHNICAL SUPPORT

If you find that you cannot determine the cause of your issue from the object-spawner log, you can always [contact SAS Technical Support](#). When you open a tracking entry, be sure to include the following:

- all the logs that encompass the time during which the issue is seen
- a clear problem description
- your [site number](#)
- your environment information (operating system, SAS version, and so on)

For more information about how to ensure that you have collected all the relevant information, see [SAS Note 57691](#), "Four tips to remember when you contact SAS Technical Support."

CONCLUSION

The object-spawner log is very helpful when you need to determine what is causing an issue with your application servers. When you experience issues with your object spawner or any application servers, the object-spawner log is the first place that you should look to investigate the cause. This paper helped you better understand what the object spawner is and how to read its logs, what might be causing an issue, and possible solutions. Remember that you can always contact SAS Technical Support to help determine the cause and solution for an issue that you encounter.

"Scientific inquiry starts with observation. The more one can see, the more one can investigate." – Martin Chalfie

REFERENCES

SAS Institute Inc. 2018. "Enable More Detailed Logging for SAS Object Spawner Troubleshooting." *SAS® 9.4 Intelligence Platform: System Administration Guide, Fourth Edition*. Cary, NC: SAS Institute Inc. Available at go.documentation.sas.com/?docsetId=bisag&docsetTarget=p1culooogaxct6jnlw9yv4wz03qwz.htm&docsetVersion=9.4&locale=en.

SAS Institute Inc. 2018. "Loggers." *SAS® 9.4 Logging: Configuration and Programming Reference, Second Edition*. Cary, NC: SAS Institute Inc. Available at go.documentation.sas.com/?docsetId=logug&docsetTarget=p12qreymyrhstun1fpqphqx i8oc2.htm&docsetVersion=9.4&locale=en.

SAS Institute Inc. 2018. "Object Spawner Messages." *SAS® 9.4 Intelligence Platform: Application Server Administration Guide*. Cary, NC: SAS Institute Inc. Available at go.documentation.sas.com/?cdcId=bicdc&cdcVersion=9.4&docsetId=biasag&docsetTarget=n21001intelplatform00srvradm.htm&locale=en.

SAS Institute Inc. 2016. SAS Note 58183. "The SAS® Workspace Server and SAS® Pooled Workspace Server fail to authenticate if the sastrust or sassrv account is an unrestricted user account." Available at support.sas.com/kb/58/183.html.

SAS Institute Inc. 2018. "Spawner Invocation Options." *SAS® 9.4 Intelligence Platform: Application Server Administration Guide*. Cary, NC: SAS Institute Inc. Available at go.documentation.sas.com/?docsetId=biasag&docsetTarget=n12027intelplatform00srvradm.htm&docsetVersion=9.4&locale=en.

ACKNOWLEDGMENTS

The author would like to thank her colleague Garrett Allred from SAS for his time, efforts, and contributions to make this paper better for customers.

RECOMMENDED READING

- *SAS® 9.4 Intelligence Platform: Application Server Administration Guide*
- *SAS® 9.4 Logging: Configuration and Programming Reference, Second Edition*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jessica Franklin
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Email: support@sas.com
Web: support.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.