

## SAS LifeCycle Listener

For SAS middle tier, it is necessary to wait for prerequisite servers to start and begin listening on their service ports. In order to start properly, many SAS web applications must connect to other servers during their initialization process.

Some of the servers include:

- SAS Web Server, if it is configured
- Database servers, including SAS Web Infrastructure Platform Data Server and third-party data servers
- SAS Metadata Server
- SAS Cache Locator, only for the server where SAS Web Infrastructure Platform is deployed
- SAS JMS Broker

If these prerequisite servers are not running, failures might occur during application initialization.

In some or all of the conditions below, it is recommended that the requisite SAS services are started before their dependent SAS web application services are started.

- 1) The SAS solution is installed on multiple machines
- 2) Has a large SAS software stack installed with multiple SAS JVMs; SAServerN\_n
- 3) When SAS multiple servers are configured to start at the same time (typically on WIN platform)
- 4) SAS Services have “Automatic” as their Start type

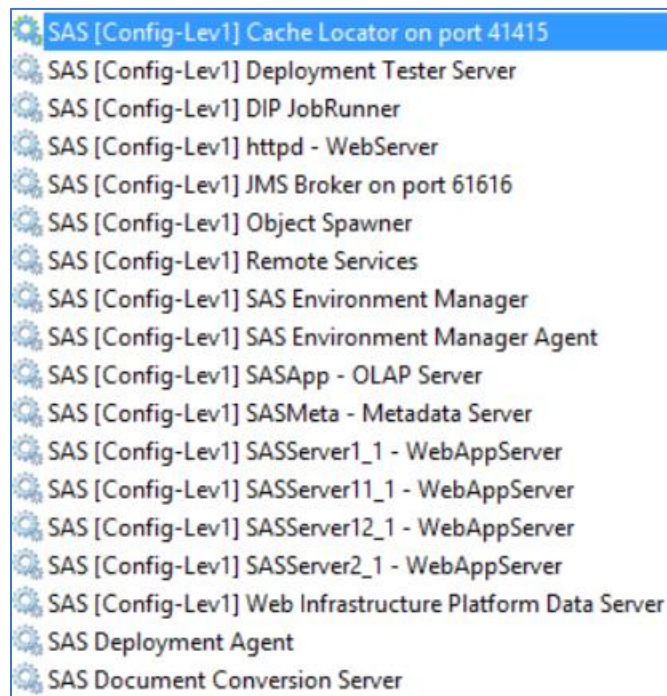


Figure 1: Sample list of SAS services on a Single Server (Aii-Lab)

## Solution

SAS introduced The LifeCycle Listener in SAS 9.4TS1M2. LifeCycle listener can force the SAS web applications services to wait for the pre-requisite servers to be started before starting the SAS web application JVMs.

## Implementation

This solution is not enabled out of the box (default), it requires manual addition to each SAS web application JVM. In other words, if you have multiple JVMs, SASServer1\_1 ... SASServerN\_n you may need to implement this solution in each one of them.

### Steps

#### Stop SAS Web Application

Navigate to SAS Services in local services and Stop all SAS Web Applications.

#### Make backup

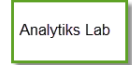
... \SAS\Config\Lev1\Web\WebAppServer\SASServer1\_1\conf\server.xml to server\_bkup\_<date>.xml

#### Edit the server.xml

In the file, locate **server** element and add

```
<Listener className="com.sas.vfabrictcsvr.startup.PrerequisiteServerListener"/>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Server port="{base.shutdown.port}" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>
  <Listener className="com.springsource.tcserver.serviceability.deploy.TcContainerDeployer"/>
  <Listener className="com.sas.vfabrictcsvr.atomikos.AtomikosLifecycleListener"/>
  <Listener className="com.sas.vfabrictcsvr.startup.PrerequisiteServerListener"/>
  <Listener accessFile="{catalina.base}/conf/jmxremote.access" address="127.0.0.1" authenticate="true" bind="sasoa.iii3.local" cla
  <GlobalNamingResources>
    <Resource auth="Container" description="User database that can be updated and saved" factory="org.apache.catalina.users.Memory
  <Resource auth="Container" description="TransactionManager" factory="com.atomikos.icatch.jta.TransactionManagerFactory" name="Trai
  <Resource auth="Container" mail.smtp.host="sasoa.iii3.local" mail.smtp.port="25" name="sas/mail/Session" type="javax.mail.Session'
  <Resource auth="Container" driverClassName="org.postgresql.Driver" factory="com.sas.vfabrictcsvr.atomikos.BeanFactory" maxPoolSize
  <Resource auth="Container" factory="com.sas.vfabrictcsvr.atomikos.BeanFactory" maxPoolSize="20" name="sas/jms/TopicConnectionFact
  <Resource auth="Container" factory="com.sas.vfabrictcsvr.atomikos.BeanFactory" maxPoolSize="20" name="sas/jms/QueueConnectionFact
  <Resource auth="Container" description="AlertQueue" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jms/AlertQue
  <Resource auth="Container" description="WorkflowQueue" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jms/Work
  <Resource auth="Container" description="AuditQueue" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jms/AuditQue
  <Resource auth="Container" description="WorkflowCommandQueue" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jr
  <Resource auth="Container" description="WorkflowEventsQueue" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jms
  <Resource auth="Container" description="WorkflowEventTopic" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jms
  <Resource auth="Container" description="NotificationEventTopic" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas
  <Resource auth="Container" description="SASPublishedEventTopic" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas
  <Resource auth="Container" description="JESSEventTopic" factory="org.apache.activemq.jndi.JNDIReferenceFactory" name="sas/jms/JFSE
  <Resource auth="Container" description="SASAuthorizationEventTopic" factory="org.apache.activemq.jndi.JND
  <Resource auth="Container" description="SASSchedulingEventTopic" factory="org.apache.activemq.jndi.JND
  <Resource auth="Container" description="scheduler.pip.resultsQueue" factory="org.apache.activemq.jndi.JND
  <Resource auth="Container" description="scheduler.pip.outputQueue" factory="org.apache.activemq.jndi.JND
  <Resource auth="Container" description="scheduler.pip.feedbackQueue" factory="org.apache.activemq.jndi.JND
  </GlobalNamingResources>
</Server>
</SAS>
```



Save and start SAS web application.

**Note:** For clustered Middle tier the process and setup is slightly different and not in scope of this article