Configuration Guidelines and Details for "HOSTNAME"

Overview

Warnings and Notices

The following issues occurred during the automated portion of your configuration and must be addressed before proceeding.

- Some of the clients in your environment may need access to the SAS/GRAPH Java applets to view graphical output generated by SAS processes. If so, add the APPLETLOC option to the sasv9_usermods.cfg file located in each SAS Application Server context directory specifying the location to be used in generating graphical output. APPLETLOC should point to a network accessible copy of the SAS/GRAPH Java applets.

  Since your deployment includes the SAS Web Infrastructure Platform, the applets can be made available (you must use the correct host and port) via:

  ```
  -APPLETLOC "http://mid-tier-server:mid-tier-port/sasweb/graph"
  ```

- You indicated that security policies could be modified by the SAS Deployment Wizard. As a result, the user HOSTNAME\sasdemo was granted the local security policy **Log on as a batch job (SeBatchLogonRight)**. This policy allows a user to be logged on by means of a batch-queue facility and is required when this account is used with SASApp - Workspace Server.

  The policy will take effect the next time HOSTNAME\sasdemo authenticates against the operating system. Therefore, if HOSTNAME\sasdemo is currently logged on to the system, a reboot will be needed for this policy to take effect.

- You indicated that security policies could be modified by the SAS Deployment Wizard. As a result, the user HOSTNAME\sasdemo was granted the local security policy **Log on as a batch job (SeBatchLogonRight)**. This policy allows a user to be logged on by means of a batch-queue facility and is required when this account is used with SASMeta - Workspace Server.

  The policy will take effect the next time HOSTNAME\sasdemo authenticates against the operating system. Therefore, if HOSTNAME\sasdemo is currently logged on to the system, a reboot will be needed for this policy to take effect.

- Because you selected to not automatically deploy your SAS Web Applications, you will need to perform some manual steps to complete the configuration of your SAS Web Applications. Once you have deployed the SAS Web Infrastructure Platform, you will need to manually load content to the SAS Content Server. Follow the instructions below to accomplish this.

SAS Management Console

SAS Management Console is required to complete many of the following steps.

Shortcut	Programs > SAS > SAS Management Console 9.2
User ID	SAS Administrator user ID
Password	Enter the password you created in the SAS Deployment Wizard.

SAS Application Servers

SASMeta - Metadata Server

Host machine	HOSTNAME.example.com
Port	8561
Log directory	C:\SAS\Config\Lev1\SASMeta\MetadataServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Execution type	Windows service SAS [Config-Lev1] SASMeta - Metadata Server
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

Notes:

- The metadata server is configured to use journaling. The metadata journal file is located at **C:\SAS\Config\Lev1\SASMeta\MetadataServer\Journal\MetadataJournal.dat**. The metadata journal file is updated more quickly than repositories and updated metadata becomes available to clients sooner. Journaling also improves the ability to recover the server and can be recovered from the state of the journal file when the metadata server is restarted.
- In the event of severe problems, the metadata server has been instructed to send alert e-mail to **admin@example.com** with the sender of the e-mail being identified as **SASMeta - Metadata Server (Config - Lev1) <admin@example.com>**.

SASApp - Connect Server

Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Expand SASApp. 3. Expand SASApp - Logical Connect Server. 4. Highlight the SASApp - Connect Server. 5. Right mouse click this server and select Validate (you can enter HOSTNAME\sasdemo or the SAS Spawned Servers account for the user ID). You should see a *Validation Successful* message.

SASApp - OLAP Server

Host machine	HOSTNAME.example.com
Port	5451
Log directory	C:\SAS\Config\Lev1\SASApp\OLAPServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Expand SASApp. 3. Highlight the SASApp - Logical OLAP Server. 4. Right mouse click this server and select Validate. You should see a *Validation Successful* message.
Execution type	Windows service SAS [Config-Lev1] SASApp - OLAP Server
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

SASApp - Stored Process Server

Host machine	HOSTNAME.example.com
Port	8601
Log directory	C:\SAS\Config\Lev1\SASApp\StoredProcessServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Expand SASApp. 3. Highlight the SASApp - Logical Stored Process Server.

4. Right mouse click this server and select Validate. You should see a *Validation Successful* message.

SASMeta - Workspace Server

Host machine	HOSTNAME.example.com
Validation steps	The SASMeta - Workspace Server should be used for only a few designated administrative tasks. It is appropriate to defer validation of this server until you have set up the SAS users who will perform these tasks. This server requires an external account and is available to only the SAS Administrators group.

Notes:

- The C:\SAS\Config\Lev1\SASMeta\WorkspaceServer\Logs directory is intended to serve as a placeholder for workspace server logs. Workspace server logs may be directed to this directory in the event that your SAS or IT administrator enables logging. The following default attributes are associated with the Logs directory:
  - Universal read, write or execute access is disabled for the directory in order to comply with default installation security requirements
  - Execution of a workspace server does not populate the Logs directory
  - The Logs directory serves as a placeholder should you enable logging for workspace servers

  Enabling workspace server logging should be carefully considered, due to the proliferation of workspace server logs, as well as the requirement to grant universal write access to the Logs directory. To enable workspace server logging, perform the following steps:
  - Enable read, write and execute access to the Logs directory
  - Modify the workspace server logconfig.xml to enable the desired levels of logging
  - Alternately, enable the logconfig.trace.xml logging file to capture workspace server diagnostic logging

SASApp - Workspace Server

Host machine	HOSTNAME.example.com
Port	8591
Log directory	C:\SAS\Config\Lev1\SASApp\WorkspaceServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Expand SASApp. 3. Highlight the SASApp - Logical Workspace Server. 4. Right mouse click this server and select Validate (you can enter HOSTNAME\sasdemo or the SAS Spawned Servers account for the user ID). You should see a *Validation Successful* message.

Notes:

- The C:\SAS\Config\Lev1\SASApp\WorkspaceServer\Logs directory is intended to serve as a placeholder for workspace server logs. Workspace server logs may be directed to this directory in the event that your SAS or IT administrator enables logging. The following default attributes are associated with the Logs directory:
  - Universal read, write or execute access is disabled for the directory in order to comply with default installation security requirements
  - Execution of a workspace server does not populate the Logs directory
  - The Logs directory serves as a placeholder should you enable logging for workspace servers

  Enabling workspace server logging should be carefully considered, due to the proliferation of workspace server logs, as well as the requirement to grant universal write access to the Logs directory. To enable workspace server logging, perform the following steps:
  - Enable read, write and execute access to the Logs directory
  - Modify the workspace server logconfig.xml to enable the desired levels of logging
  - Alternately, enable the logconfig.trace.xml logging file to capture workspace server diagnostic logging

SAS BI Report Services Workspace Configuration

Query Cache Library	Name ="SASApp - wrstemp" 1. In SAS Management Console, on the Plug-ins tab, select the "+" to expand the Data Library Manager node. 2. Select "+" to expand Libraries. 3. Select the library listed above. 4. Right mouse click this library and select Properties. 5. Click the Options tab and locate the path to the library in the Selected items text box.

	Review the documentation provided to determine if permissions need to be modified for the library or directory to which the library points. 1. For details, view the http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92 and refer to the document called ***SAS 9.2 Intelligence Platform: Web Application Administration Guide***. 2. See the section called ***Improving the Performance of SAS Web Report Studio***. 3. This section will contain a list of *Suggestions for Improving the Performance of SAS Web Report Studio.* Go to the information titled ***"Using the Query Cache"***. 4. Refer to the subsection called ***Manage Host Access to the Query Cache Library*** for information regarding permissions.
Distribution Library	Name ="SASApp - wrsdist" 1. In SAS Management Console, on the Plug-ins tab, select the "+" to expand the Data Library Manager node. 2. Select "+" to expand Libraries. 3. Select the library listed above. 4. Right mouse click this library and select Properties. 5. Click the Options tab and locate the path to the library in the Selected items text box. Review the documentation provided to determine if permissions need to be modified for the library or directory to which the library points. 1. For details, view the http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92 and refer to the document called ***SAS 9.2 Intelligence Platform: Web Application Administration Guide***. 2. See the section called ***Pre-generated Reports From SAS Web Report Studio***. 3. Refer to the subsection called ***Verifying Permissions for the Distribution Library*** for information regarding permissions.
Validation steps	1. In SAS Management Console, on the Plug-ins tab, select "+" to expand the Data Library Manager node. 2. Verify the libraries named above are listed under the Data Library Manager. 3. In SAS Management Console, on the Plug-ins tab, select "+" to expand the Application Management node. 4. Select "+" to expand the Configuration Manager 5. Highlight BI Rep Svc Wkspace Config 4.2 6. Right mouse click and select Properties. 7. Click the Settings tab. 8. Validate the query cache library and the distribution library. They should match the libraries listed under Data Libraries.

SAS BI Report Services 4.2

Log directory	C:\SAS\Config\Lev1\Applications\HOSTNAMEReportServices4.2\Logs For more details about how to modify your log format and to see what logging entails, see the *SAS Intelligence Platform: Web Application Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Java Batch Server	Name ="SASApp - BRM Java Batch Server"
Configuration	Configuration Directory = "C:\SAS\Config\Lev1\Applications\HOSTNAMEReportServices4.2"
Validation steps	**Java Batch Server** 1. In SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Select "+" to expand SASApp. 3. Select "+" to expand the SASApp - Logical SAS Java Batch Server. 4. Verify the existence of the java batch server(s) referenced above. 5. Right mouse click this server and select Properties. 6. Click the Options tab. 7. Validate the command line by examining the parameter values and verifying file locations and the repository name. **Local Services** 1. In SAS Management Console, on the Plug-ins tab, select "+" to expand the Foundation Services Manager node. 2. Select "+" to expand the HOSTNAMEReportServices4.2 Local Services Definition. 3. Select "+" to expand Core. 4. Select the Logging Service. 5. Right mouse button and choose Properties. 6. Click the Service Configuration tab. 7. Click the Configuration button. 8. Verify the logging contexts. 9. Refer to the logging documentation at the link specified above for details. **outputgen.ini file** 1. The file outputgen.ini may need to be modified. This file is located in the install folder C:\Program Files\SAS\HOSTNAMEReportServices\4.2. 2. The following default settings for Java command line options appear in this file: java.net.preferIPv4Stack=true and java.net.preferIPv6Addresses=false. 3. If the system is configured to use IPv6, these values must be changed to: java.net.preferIPv4Stack=false and java.net.preferIPv6Addresses=true.

SASApp - Pooled Workspace Server

Host machine	HOSTNAME.example.com
Port	8701
Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Expand SASApp. 3. Highlight the SASApp - Logical Pooled Workspace Server. 4. Right mouse click this server and select Validate. You should see a *Validation Successful* message.

SASMeta - SAS DATA Step Batch Server

Host machine	HOSTNAME.example.com
Log directory	C:\SAS\Config\Lev1\SASMeta\BatchServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. Select this link: C:\SAS\Config\Lev1\SASMeta\BatchServer\sasbatch.bat 2. If SAS starts successfully, the DATA step batch server has been validated.

SASApp - SAS DATA Step Batch Server

Host machine	HOSTNAME.example.com
Log directory	C:\SAS\Config\Lev1\SASApp\BatchServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. Select this link: C:\SAS\Config\Lev1\SASApp\BatchServer\sasbatch.bat 2. If SAS starts successfully, the DATA step batch server has been validated.

SAS Spawners

Object Spawner

Host machine	HOSTNAME.example.com
Port	8581
Log directory	C:\SAS\Config\Lev1\ObjectSpawner\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Execution type	Windows service SAS [Config-Lev1] Object Spawner
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

Connect Spawner

Host machine	HOSTNAME.example.com
Ports	7551
Log directory	C:\SAS\Config\Lev1\SASApp\ConnectServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.

Execution type	Windows service SAS [Config-Lev1] Connect Spawner
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

SAS Table Server

Host machine	HOSTNAME.example.com
Port	2171
Log directory	C:\SAS\Config\Lev1\SASTS\TableServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Execution type	Windows service SAS [Config-Lev1] SASTS - Table Server
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

SAS/SHARE Server

Host machine	HOSTNAME.example.com
Port	8551
Log directory	C:\SAS\Config\Lev1\ShareServer\Logs For more details about the initial logging configuration and how to modify it, see "Administering Logging for SAS Servers" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Execution type	Windows service SAS [Config-Lev1] Share Server
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

SAS Deployment Tester Server

Host machine	HOSTNAME.example.com
Port	10021
Execution type	Windows service SAS [Config-Lev1] Deployment Tester Server
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1
Validation steps	1. In SAS Management Console, on the Plug-ins tab, select "+" to expand the Application Management node. 2. Select "+" to expand Deployment Tester 3. Highlight the host machine: HOSTNAME.example.com 4. Right mouse click and select Run All Test Suites. 5. Select "+" to expand the Results. 6. Validate the results for each test suite. All tests should pass.

Notes:

- Deployment Tester Server is used to run tests on SAS products and servers to validate functionality and proper operation. The Deployment Tester Server in conjunction with the Deployment Tester SMC plug-in allows for tests to be executed on remote systems from a single SMC instance. Select Help on Deployment Tester in the SMC Help menu for more information.

Operating System Services Scheduling Server

Host machine	HOSTNAME.example.com
Port	8451

Log directory	C:\SAS\Config\Lev1\SchedulingServer\Logs
Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Highlight Operating System Services. 3. Right mouse click this server and select Validate (you can enter HOSTNAME\sasdemo or the SAS Spawned Servers account for the user ID). You should see a *Validation Successful* message.

Remote Services

Host machine	HOSTNAME.example.com
Port	5091
Execution type	Windows service SAS [Config-Lev1] Remote Services
Shortcuts	Programs > SAS > SAS Configuration > Config - Lev1

Notes:

- A shortcut has been created to start SAS Remote Services and can be found in your Start Menu at: Programs > SAS > SAS Configuration > Config - Lev1 > Remote Services - Start
- A shortcut has been created to stop SAS Remote Services and can be found in your Start Menu at: Programs > SAS > SAS Configuration > Config - Lev1 > Remote Services - Stop

Web Application Server

Host machine	HOSTNAME.example.com
Application Server Vendor	IBM
Application Server Name	WebSphere
WebSphere Version	6.1.0.23
WebSphere Fix Pack Level	23
WebSphere Java Version	"1.5.0"
WebSphere Java Runtime	Java(TM) 2 Runtime Environment, Standard Edition (build pwa64devifx-20080907 (SR8a + IZ29767 + IZ30684 + IZ31214 + IZ31213))
WebSphere Installation Directory	C:\Program Files\IBM\WebSphere\AppServer
WebSphere Admin Console URL	http://HOSTNAME:9060/ibm/console
WebSphere Admin Console Http Port	9060
WebSphere Admin Console Https Port	9043
WebSphere Cell Name	HOSTNAMECell01
WebSphere Deployment Manager Profile Name	Dmgr01
WebSphere Dmgr Node Name	HOSTNAMECellManager01
WebSphere Dmgr Server Logs Directory	C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\logs\dmgr
WebSphere Managed Node Profile Name	AppSrv01
WebSphere Managed Node Name	HOSTNAMENode01
WebSphere Managed Node Logs Directory	C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\logs

==========	==========
Application Server Name	server1
Application Server Id	server
Application Server Http Port	9080
Application Server Log Directory	C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\server1
Application Server JVM Options	-Xms1024m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:optavgpause -Djavax.management.builder.initial= -Dcom.sun.management.jmxremote -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -Dsas.container.identifier=websphere -Dsas.auto.publish.port=9080 -Dcom.sas.services.logging.disableRemoteList=true -Dcom.sas.services.logging.disableRemoteLogging=true -Djava.net.preferIPv4Stack=true -Djava.net.preferIPv6Addresses=false -Dmulticast_udp_ip_ttl=0 -Dmulticast.address=235.1.1.1 -Dmulticast.port=31001 -Dcom.sas.log.config.url=file:///C:/SAS/Config/Lev1/Web/Common/LogConfig/

Configuring your WebSphere Application Server Environment

SPECIAL NOTE ABOUT MANUAL CONFIGURATION: Since you have requested that the WebSphere environment **NOT** be auto-configured, **you are responsible for manually configuring WebSphere using the instructions outlined below**.
Following these instructions should provide a basic WebSphere configuration that uses SAS authentication and unsecure connnections.
Where possible, you may "cut-and-paste" names, values and commands from these instructions to appropriate fields in the WAS Admin Console or operating system command-lines.
Be careful to note when leading and trailing "quotes" must be ignored when transferring values to the WAS Admin Console or to operating system command-lines.

If you need to setup a more sophisticated topology such as web server authentication or single sign-on, please see the *SAS 9.2 Intelligence Platform: Security Administration Guide.*
For reverse proxy server and secure socket connections, see the *SAS 9.2 Intelligence Platform: Web Application Administration Guide.*

SPECIAL NOTE ABOUT MANUAL DEPLOYMENT: Since auto-deployment of the SAS web applications was **NOT** performed during execution of the SAS Deployment Wizard, **you are responsible for manually deploying all of the SAS web applications.**
EAR files for each of the configured SAS web applications have been constructed and are located in the "C:\SAS\Config\Lev1\Web\Staging" directory.
You can use the WAS Admin Console to manually deploy these EAR files. See "Deployment for application..." for instructions on how to perform this function.
Once manual deployment of all SAS web applications has been completed, you must start all SAS web applications using the WAS Admin Console by selecting all applications and clicking Start.

Create WebSphere Profiles

Create the WebSphere dmgr profile to manage SAS application server node(s)

The following is the general overview of the steps required to create the WebSphere dmgr "Dmgr01" profile:

- Create dmgr profile "Dmgr01"
- If you are configuring WAS manually, you will need to start the dmgr at this point, and logon to the WAS Admin Console
- Set dmgr Http Ports
- Set Http ports for the admin_host virtualhosts alias
- Set Generic JVM Arguments for dmgr
- If SOAP is used for dmgr-to-nodeagent communication, update dmgr SOAP Read Timeout Property
- If auto-configuration for WAS is being performed, dmgr will now be started

Detailed Instructions to Create the WebSphere "Dmgr01" dmgr Profile

1. **Create the "Dmgr01" profile** using the "manageprofiles" command to construct a new cell which will contain the dmgr server.

   Command to create dmgr profile: "C:\Program Files\IBM\WebSphere\AppServer\bin\manageprofiles.bat" -create -profileName "Dmgr01" -profilePath "C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01" -templatePath "C:\Program Files\IBM\WebSphere\AppServer\profileTemplates\dmgr" -nodeName "HOSTNAMECellManager01" -cellName "HOSTNAMECell01" -defaultPorts -isDefault -winserviceCheck true -winserviceStartupType automatic

2. **Start the dmgr manager** using the "startManager" command.
   Command to start the dmgr server: "C:\Program Files\IBM\WebSphere\AppServer\bin\startManager.bat" -profileName Dmgr01

3. **Logon to the WAS Admin Console** using http://HOSTNAME.example.com:9060/ibm/console, when the dmgr has started successfully.

4. **Set dmgr Ports** to the following values by either editing the "C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\config\cells\HOSTNAMECell01\nodes\HOSTNAMECellManager01\serverindex.xml" file or using the WAS Admin Console "System administration->Deployment manager->ports" navigation to perform this task.

- Set dmgr WC_adminhost port to 9060
- Set dmgr WC_adminhost_secure port to 9043
- Set dmgr BOOTSTRAP_ADDRESS port to 9809
- Set dmgr SOAP_CONNECTOR_ADDRESS port to 8879

5. **Set the Http ports for the admin_host virtualhost** alias to match port 9060 for dmgr to the following values by either editing the "C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\config\cells\HOSTNAMECell01\virtualhosts.xml file or using the WAS Admin Console "Environment->Virtual Hosts->admin_host->Host Aliases" navigation to perform this task.

   - Set the admin_host alias for "*" to the Http port 9060 for dmgr
   - Set the admin_host alias for "*" to the Https port 9043 for dmgr

6. **Set the dmgr JVM Initial Heap Size and Maximum Heap Size**. From the WAS Admin Console, navigate through the "System administration->Deployment manager->Java and Process Management->Process Definition->Java Virtual Machine" menus.

   - Set dmgr "Initial Heap Size" to 512
   - Set dmgr "Maximum Heap Size" to 768
   - Set dmgr server Generic JVM Arguments to "-Xms512m -Xmx768m" (**NOTE:** remember to exclude the beginning and ending quotes.)

7. **"Bounce" the dmgr manager** to allow the server updates to take effect using the "stopManager" command to stop the dmgr manager and the "startManager" command to restart it.
   Command to stop the dmgr server: "C:\Program Files\IBM\WebSphere\AppServer\bin\stopManager.bat" -profileName "Dmgr01"
   Command to start the dmgr server: "C:\Program Files\IBM\WebSphere\AppServer\bin\startManager.bat" -profileName "Dmgr01"

At this point, the dmgr profile "Dmgr01" should be created.

Create the WebSphere managed node profile for the nodeagent and SAS application servers.

The following is the general overview of the steps required to create the WebSphere custom managed node "AppSrv01" profile:

- If auto-configuration for WAS is being performed, it is assumed that the dmgr is started
- Create the App Server managed profile "AppSrv01"
- Set nodeagent RMI and SOAP Ports, if necessary
- Set Generic JVM Arguments for nodeagent
- If SOAP is used for dmgr-to-nodeagent communications, update dmgr SOAP Read Timeout Property
- Federate the new managed node "HOSTNAMENode01" into the dmgr "HOSTNAMECell01" cell
- If on Windows, ensure the nodeagent is stopped and create a Windows Service for the nodeagent
- Synchronize the managed node with the dmgr before starting the nodeagent using the syncNode command
- Ensure the nodeagent is started for the newly federated node

Detailed Instructions to Create the WebSphere "AppSrv01" Managed Node Profile

1. **Create the App Server managed profile "AppSrv01"** using the "manageprofiles" command.
   Command to create the app server profile: "C:\Program Files\IBM\WebSphere\AppServer\bin\manageprofiles.bat" -create -profileName "AppSrv01" -profilePath "C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01" -templatePath "C:\Program Files\IBM\WebSphere\AppServer\profileTemplates\managed" -nodeName "HOSTNAMENode01" -cellName "AppSrv01Cell" -nodeDefaultPorts -federateLater true -hostName HOSTNAME.example.com -winserviceCheck true -winserviceStartupType automatic

2. **Federate the new "HOSTNAMENode01" node to the dmgr cell** using the "addNode" command.
   Command to federate the new node: "C:\Program Files\IBM\WebSphere\AppServer\bin\addNode.bat" "HOSTNAME" 8879 -conntype SOAP -profileName "AppSrv01" -includeapps -noagent

3. **Set nodeagent Ports** to the following values by either editing the "C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\HOSTNAMECell01\nodes\HOSTNAMENode01\serverindex.xml" file or using the WAS Admin Console "System administration->Node agents->nodeagent for "HOSTNAMENode01"->ports" navigation to perform this task.

   - Set nodeagent BOOTSTRAP_ADDRESS port to 2809
   - Set nodeagent SOAP_CONNECTOR_ADDRESS port to 8880

4. **Set the nodeagent JVM Initial Heap Size and Maximum Heap Size**. From the WAS Admin Console, navigate through the "System administration->Node agents->nodeagent for Node "HOSTNAMENode01"->Java and Process Management->Process Definition->Java Virtual Machine" menus.

   - Set nodeagent "Initial Heap Size" to 50
   - Set nodeagent "Maximum Heap Size" to 256
   - Set nodeagent server Generic JVM Arguments to "-Xms50m -Xmx256m" (**NOTE:** remember to exclude the beginning and ending quotes.)

5. **Synchronize the nodeagent and dmgr repositories** using the "syncNode" command.
   Command to synchronize the nodeagent with the dmgr: "C:\Program Files\IBM\WebSphere\AppServer\bin\syncNode.bat" "HOSTNAME" 8879 -conntype SOAP -profileName "AppSrv01"

6. **Create a Windows Service for the nodeagent** using the "WASService" command.
   Command to create a Windows Service for the nodeagent: "C:\Program Files\IBM\WebSphere\AppServer\bin\WASService" -add HOSTNAMENode01 -servername nodeagent -restart true -startType automatic -profilePath "C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01"

7. **Start the nodeagent** using the "startNode" command.
   Command to start the nodeagent: "C:\Program Files\IBM\WebSphere\AppServer\bin\startNode.bat" -profileName "AppSrv01"

Creation of WAS Dmgr profile should be complete at this point. You should be able to run the "manageprofiles" command with the "-listProfiles" parameter to verify that the "Dmgr01" profile exists. In addition, you should see a Java process running for the "dmgr" server. You should also be able logon to the WAS Admin Console using the URL: http://HOSTNAME:9060/ibm/console.

Creation of WAS managed node profile should be complete at this point. You should be able to run the "manageprofiles" command with the "-listProfiles" parameter to verify that the "AppSrv01" profile exists. In addition, you should see a Java processes running for the "nodeagent" server.

Create Application Server "server1" Overview

- Create Application Server "server1"
- Set server1 Http Ports
- Set Http ports for the default_host virtualhosts alias
- Set Generic JVM Arguments for "server1"
- Enable Java 2 Security for "server1", if restrictive policy support for access to local resources is desired.
- Set the required Web Container custom properties for "server1"
- Create the Service Integration Bus for "server1"
- Add server "server1" as a member on the Service Integration Bus
- Perform a Full Resynchronize of the dmgr and nodeagent

Create Application Server Detailed Instructions

1. **Create a new application server "server1"** using the WAS Admin Console.
   - Use Servers->Application servers->New navigation to create a new application server.
   - Ensure that HOSTNAMENode01 node has been selected in the "Select node" drop-down menu.
   - Enter "server1" in required "Server name" field.
   - Select the "default" server template radio button.
   - Check the "Generate unique ports" checkbox.
   - Save your work after reviewing the summary page.

2. **Set Http Ports for "server1"** to the following values by either editing the "C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\HOSTNAMECell01\nodes\HOSTNAMENode01\serverindex.xml" file or using the WAS Admin Console by navigating to Application servers->server1->Ports to perform this function.
   - Set server1 WC_defaulthost port to 9080
   - Set server1 WC_defaulthost_secure port to 9443

3. **Set the Http ports for the default_host virtualhost alias to match the Http ports for "server1".** The following Http port values should be set by either editing the "C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\config\cells\HOSTNAMECell01\virtualhosts.xml file or using the WAS Admin Console by navigating to Environment->Virtual Hosts->default_host->Host Aliases to perform this task.
   - Set the default_host alias for "*" to match the Http port 9080 for server1
   - Set the default_host alias for "*" to match the Https port 9443 for server1

4. **Set Generic JVM Arguments for server "server1"** using the WAS Admin Console. after navigating to Application servers->server1->Java and Process Management->Process Definition->Java Virtual Machine.
   server1 Generic JVM Arguments (exclude the beginning and ending "quotes"):
   - Set server1 "Initial Heap Size" to 1024
   - Set server1 "Maximum Heap Size" to 2048
   - Set server1 Generic JVM Arguments to: "-Xms1024m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:optavgpause -Djavax.management.builder.initial= -Dcom.sun.management.jmxremote -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -Dsas.container.identifier=websphere -Dsas.auto.publish.port=9080 -Dcom.sas.services.logging.disableRemoteList=true -Dcom.sas.services.logging.disableRemoteLogging=true -Djava.net.preferIPv4Stack=true -Djava.net.preferIPv6Addresses=false -Dmulticast_udp_ip_ttl=0 -Dmulticast.address=235.1.1.1 -Dmulticast.port=31001 -Dcom.sas.log.config.url=file:///C:/SAS/Config/Lev1/Web/Common/LogConfig/ " (**NOTE:** remember to exclude the beginning and ending quotes.)

5. **Disable Java 2 Security for server "server1"** using the WAS Admin Console.
   Using the "Secure administration, applications, and infrastructure" menu in the WAS Admin Console, disable Java 2 Security by unchecking the "Java 2 security" checkbox.

6. **Set the required Web Container custom properties for server "server1"** using the WAS Admin Console.
   Using the Application servers->server1->Web Container Settings->Web container->Custom Properties menu in the WAS Admin Console, there are two custom properties that must be set for proper SAS web application function.
   - Create a new property with a name of "prependSlashToResource" and a value of "true" (without the "quotes").
   - Create a new property with a name of "com.ibm.ws.webcontainer.channelwritetype" and a value of "sync" (again, without the "quotes").

7. **Create the Service Integration Bus for server "server1"** using the WAS Admin Console.
   Using the Service integration menu in the WAS Admin Console, create a new bus with the name "SAS Messaging Bus" (without the "quotes").

8. **Add server "server1" as a member on the Service Integration Bus** "SAS Messaging Bus" created in the step above using the WAS Admin Console.
   From the the Service integration->Buses menu in the WAS Admin Console, select "SAS Messaging Bus", use the Bus members link to add "HOSTNAMENode01:server1" as a new member of the bus. The "Server" radio button should be selected with the "HOSTNAMENode01:server1" (without the "quotes") drop-down menu item. Choose all the pre-selected and pre-filled defaults on the remaining definition pages.

9. **After saving your work, perform a Full Resynchronize** of the dmgr and nodeagent using the WAS Admin Console.
   From the the System administration menu, choose the "HOSTNAMENode01" and perform a Full Resynchronize to ensure all new server definitions are propagated from the master repository to the nodeagent.

   Creation of server "server1" should now be complete. The base JVM and Web Container configuration for the new "server1" has been completed. All remaining configuration steps will use these base server definitions to complete the customization of the J2EE server functions for the SAS web applications.

Create a Mail Session named "SASMailSession" for server "server1"

Use the Resources->Mail->Mail sessions menu on the WAS Admin Console to create Mail Session "SASMailSession"

1. Before clicking New, be sure the scope is set to "Node=HOSTNAMENode01, Server=server1" for the new Mail Session definition.
2. On the new Mail Session defintion page, the Provider drop-down menu can be left set to the WebSphere default "Built-in Mail Provider".
3. Complete the required input fields as follows:
   - Enter "SASMailSession" (without the "quotes") for the Mail Session Name.
   - Set the JNDI name to "sas/mail/Session" (without the "quotes").
   - Set the Mail transport host to "mailhost.example.com" (without the "quotes").
   - Set the Mail transport protocol to "smtp" (without the "quotes").
   - If the SMTP Mail tranport host requires authentication, a valid Mail transport user ID and Mail transport password must be entered.
   - **All remaining fields on the page should be remain unaltered** - empty or pre-filled default values for the remaining input fields are adequate.

Values for the Create Mail Session	
Scope:	Node=HOSTNAMENode01, Server=server1
Mail Session Name:	SASMailSession
JNDI Name:	sas/mail/Session
SMTP Host Name:	mailhost.example.com
Mail Debug:	false
Mail Transport Protocol:	smtp
Mail SMTP Host Requires Authentication:	false

Add the JAAS Alias "PFS" for JAAS Application Login Module "com.sas.services.security.login.OMILoginModule" to the cell configuration.

Use the WAS Admin Console to add JAAS Login Module Alias "PFS" for the JAAS Login Module "com.sas.services.security.login.OMILoginModule" to the Application logins at the cell level.
NOTE: surrounding "quotes" for any of the following values should **NOT** be entered in the input fields!

1. Navigate to Security->Secure administration, applications, and infrastructure->Java Authentication and Authorization Service->Application logins.
2. If the "PFS" alias name does not already exist, create a new "PFS" Alias (without the "quotes").
3. Whether you have just created a new "PFS" Alias, or it was already present, click the "PFS" Alias link.
4. Click the "JAAS login modules" link.
5. On the JAAS login modules page, create a new Module class name with a value of "com.sas.services.security.login.OMILoginModule" (without the "quotes").
6. Be sure to select "sufficient" from the Authentication strategy drop-down menu.
7. Click the new JAAS Module class name link for "com.sas.services.security.login.OMILoginModule" on the JAAS login modules page.
8. Select the "Custom properties" link for "com.sas.services.security.login.OMILoginModule".
9. Use the following **Custom Properties for JAAS Login Module "com.sas.services.security.login.OMILoginModule"** table and enter the name-value pairs for each row following the **"Custom Property Name:"** and **"Custom Property Value:"** headings as a custom property for the JAAS Login Module.
10. Be sure to save your work and perform a full resynchronization.

Values for JAAS Login Module general settings:	
JAAS Application logins alias:	PFS
JAAS Module Class name:	com.sas.services.security.login.OMILoginModule
JAAS Authentication strategy:	sufficient

Custom Properties for JAAS Login Module "com.sas.services.security.login.OMILoginModule"	
Custom Property	**Custom Property Value:**

Name:	
host	HOSTNAME.example.com
port	8561
repository	Foundation
domain	DefaultAuth
debug	false
trusteduser	Enter the SAS trusted user ID
trustedpw	Enter the **SAS-encoded password** for the SAS trusted user ID. (Use this SAS program to encode the value: PROC PWENCODE IN='password'; RUN;)

Add the JAAS Alias "PFS" for JAAS Application Login Module "com.sas.services.security.login.TrustedLoginModule" to the cell configuration.

Use the WAS Admin Console to add JAAS Login Module Alias "PFS" for the JAAS Login Module "com.sas.services.security.login.TrustedLoginModule" to the Application logins at the cell level.
NOTE: surrounding "quotes" for any of the following values should **NOT** be entered in the input fields!

1. Navigate to Security->Secure administration, applications, and infrastructure->Java Authentication and Authorization Service->Application logins.
2. If the "PFS" alias name does not already exist, create a new "PFS" Alias (without the "quotes").
3. Whether you have just created a new "PFS" Alias, or it was already present, click the "PFS" Alias link.
4. Click the "JAAS login modules" link.
5. On the JAAS login modules page, create a new Module class name with a value of "com.sas.services.security.login.TrustedLoginModule" (without the "quotes").
6. Be sure to select "optional" from the Authentication strategy drop-down menu.
7. Click the new JAAS Module class name link for "com.sas.services.security.login.TrustedLoginModule" on the JAAS login modules page.
8. Select the "Custom properties" link for "com.sas.services.security.login.TrustedLoginModule".
9. Use the following **Custom Properties for JAAS Login Module "com.sas.services.security.login.TrustedLoginModule"** table and enter the name-value pairs for each row following the **"Custom Property Name:"** and **"Custom Property Value:"** headings as a custom property for the JAAS Login Module.
10. Be sure to save your work and perform a full resynchronization.

Values for JAAS Login Module general settings:	
JAAS Application logins alias:	PFS
JAAS Module Class name:	com.sas.services.security.login.TrustedLoginModule
JAAS Authentication strategy:	optional

Custom Properties for JAAS Login Module "com.sas.services.security.login.TrustedLoginModule"	
Custom Property Name:	**Custom Property Value:**
host	HOSTNAME.example.com
port	8561
repository	Foundation
domain	DefaultAuth
aliasdomain	MidtierInternal
debug	false
trusteduser	Enter the SAS trusted user ID
trustedpw	Enter the **SAS-encoded password** for the SAS trusted user ID. (Use this SAS program to encode the value: PROC PWENCODE IN='password'; RUN;)

Add the JAAS Alias "UsernamePassword" for JAAS Application Login Module "com.platform.SASLogin.UsernamePasswordLogin" to the cell configuration.

Use the WAS Admin Console to add JAAS Login Module Alias "UsernamePassword" for the JAAS Login Module "com.platform.SASLogin.UsernamePasswordLogin" to the Application logins at the cell level.
NOTE: surrounding "quotes" for any of the following values should **NOT** be entered in the input fields!

1. Navigate to Security->Secure administration, applications, and infrastructure->Java Authentication and Authorization Service->Application logins.
2. If the "UsernamePassword" alias name does not already exist, create a new "UsernamePassword" Alias (without the "quotes").
3. Whether you have just created a new "UsernamePassword" Alias, or it was already present, click the "UsernamePassword" Alias link.
4. Click the "JAAS login modules" link.
5. On the JAAS login modules page, create a new Module class name with a value of "com.platform.SASLogin.UsernamePasswordLogin" (without the "quotes").
6. Be sure to select "required" from the Authentication strategy drop-down menu.
7. Click the new JAAS Module class name link for "com.platform.SASLogin.UsernamePasswordLogin" on the JAAS login modules page.
8. Select the "Custom properties" link for "com.platform.SASLogin.UsernamePasswordLogin".

9. Use the following **Custom Properties for JAAS Login Module "com.platform.SASLogin.UsernamePasswordLogin"** table and enter the name-value pairs for each row following the **"Custom Property Name:"** and **"Custom Property Value:"** headings as a custom property for the JAAS Login Module.
10. Be sure to save your work and perform a full resynchronization.

Values for JAAS Login Module general settings:	
JAAS Application logins alias:	UsernamePassword
JAAS Module Class name:	com.platform.SASLogin.UsernamePasswordLogin
JAAS Authentication strategy:	required

Custom Properties for JAAS Login Module "com.platform.SASLogin.UsernamePasswordLogin"	
Custom Property Name:	**Custom Property Value:**
debug	false

Add the JAAS Alias "SCS" for JAAS Application Login Module "com.sas.services.security.login.OMILoginModule" to the cell configuration.

Use the WAS Admin Console to add JAAS Login Module Alias "SCS" for the JAAS Login Module "com.sas.services.security.login.OMILoginModule" to the Application logins at the cell level.
NOTE: surrounding "quotes" for any of the following values should **NOT** be entered in the input fields!

1. Navigate to Security->Secure administration, applications, and infrastructure->Java Authentication and Authorization Service->Application logins.
2. If the "SCS" alias name does not already exist, create a new "SCS" Alias (without the "quotes").
3. Whether you have just created a new "SCS" Alias, or it was already present, click the "SCS" Alias link.
4. Click the "JAAS login modules" link.
5. On the JAAS login modules page, create a new Module class name with a value of "com.sas.services.security.login.OMILoginModule" (without the "quotes").
6. Be sure to select "required" from the Authentication strategy drop-down menu.
7. Click the new JAAS Module class name link for "com.sas.services.security.login.OMILoginModule" on the JAAS login modules page.
8. Select the "Custom properties" link for "com.sas.services.security.login.OMILoginModule".
9. Use the following **Custom Properties for JAAS Login Module "com.sas.services.security.login.OMILoginModule"** table and enter the name-value pairs for each row following the **"Custom Property Name:"** and **"Custom Property Value:"** headings as a custom property for the JAAS Login Module.
10. Be sure to save your work and perform a full resynchronization.

Values for JAAS Login Module general settings:	
JAAS Application logins alias:	SCS
JAAS Module Class name:	com.sas.services.security.login.OMILoginModule
JAAS Authentication strategy:	required

Custom Properties for JAAS Login Module "com.sas.services.security.login.OMILoginModule"	
Custom Property Name:	**Custom Property Value:**
host	HOSTNAME.example.com
port	8561
repository	Foundation
domain	DefaultAuth
debug	false
holdopenconnection	true
trusteduser	Enter the SAS trusted user ID
trustedpw	Enter the **SAS-encoded password** for the SAS trusted user ID. (Use this SAS program to encode the value: PROC PWENCODE IN='password'; RUN;)

Starting WebSphere Application Server "server1"

If you haven't already done so, it's a good idea to perform a Full Resynchronize using the WAS Admin Console before starting server "server1".
This can be done from the the System administration menu, choose the "HOSTNAMENode01" and perform a Full Resynchronize to ensure any new server definitions are propagated from the master repository to the nodeagent.

Start the server "server1" using the "startServer" command.
Command to start server "server1": "C:\Program Files\IBM\WebSphere\AppServer\bin\startServer.bat" "server1" -profileName "AppSrv01"

Server "server1" should now be started. You should see a process running representing the JVM for this server.

Create a JMS Topic Connection Factory "SASTopicConnectionFactory" at the "CELL" level.

Create a new JMS Topic Connection Factory "SASTopicConnectionFactory" definition at scope "HOSTNAMECell01" level.

1. Navigate to Resources->JMS->Topic connection factories on the WAS Admin Console.
2. Ensure the Scope is set to "Cell=HOSTNAMECell01" before clicking New.
3. Be sure the "Default messaging provider" radio button is selected.
4. There are 3 required fields (marked with "asterisks") on the "Configuration" page:
   - Enter "SASTopicConnectionFactory" (without the "quotes") into the required "Name" field.
   - Enter "sas/jms/TopicConnectionFactory" (without the "quotes") into the required "JNDI name" field.
   - Select the "SAS Messaging Bus" from the required "Bus name" drop-down menu.
5. Click the "Connection pool properties" link to set the following connection pool settings for this JMS Connection Factory:
   - Connection timeout: 180
   - Maximum connections: 10
   - Minimum connections: 1
   - Reap time: 180
   - Unused timeout: 1800
   - Aged timeout: 0
   - Purge policy: EntirePool
6. **All remaining fields on the page should be remain unaltered** - empty or pre-filled default values are for the remaining input fields are adequate.
7. Be sure to Save your work.

You should perform a Full Resynchronize to ensure the new JMS Topic Connection Factory definition is propagated from the dmgr to the nodeagent.

Create a JMS Queue Connection Factory "SASQueueConnectionFactory" at the "CELL" level.

Create a new JMS Queue Connection Factory "SASQueueConnectionFactory" definition at scope "HOSTNAMECell01" level.

1. Navigate to Resources->JMS->Queue connection factories on the WAS Admin Console.
2. Ensure the Scope is set to "Cell=HOSTNAMECell01" before clicking New.
3. Be sure the "Default messaging provider" radio button is selected.
4. There are 3 required fields (marked with "asterisks") on the "Configuration" page:
   - Enter "SASQueueConnectionFactory" (without the "quotes") into the required "Name" field.
   - Enter "sas/jms/QueueConnectionFactory" (without the "quotes") into the required "JNDI name" field.
   - Select the "SAS Messaging Bus" from the required "Bus name" drop-down menu.
5. Click the "Connection pool properties" link to set the following connection pool settings for this JMS Connection Factory:
   - Connection timeout: 180
   - Maximum connections: 10
   - Minimum connections: 1
   - Reap time: 180
   - Unused timeout: 1800
   - Aged timeout: 0
   - Purge policy: EntirePool
6. **All remaining fields on the page should be remain unaltered** - empty or pre-filled default values are for the remaining input fields are adequate.
7. Be sure to Save your work.

You should perform a Full Resynchronize to ensure the new JMS Queue Connection Factory definition is propagated from the dmgr to the nodeagent.

Create a JMS Queue "AlertQueue" and Service Integration Bus Destination "AlertQueue" at the "CELL" level.

Create a new JMS Queue "AlertQueue" and Service Integration Bus Destination "AlertQueue" at scope "HOSTNAMECell01" on SIBus "SAS Messaging Bus".

1. Create the Service Integration Bus Queue destination "AlertQueue", if it doesn't already exist.
2. Navigate to Service integration->Buses->, click the "SAS Messaging Bus" link, then click "Destinations".
3. Click New and ensure the Queue radio button is selected.
4. Enter "AlertQueue" (without the "quotes") in the required "Identifier" field.
5. On the Assign the queue to a bus member page, select the entry for "Node=HOSTNAMENode01:Server=server1" from the Bus member drop-down menu.
6. Next, create the JMS Queue with a Name of "AlertQueue" to be associated with the destination just created.
7. Navigate to "Resources->JMS->Queues" menu on the WAS Admin Console.
8. On the Queues page, be sure the scope setting "Cell=HOSTNAMECell01" is selected from the drop-down menu and click New.
9. Make sure the "Default messaging provider" radio button is selected and click Ok.
10. For the input fields on the Queues->Default messaging provider->New configuration page, complete the following fields:
    - Name: AlertQueue
    - JNDI Name: sas/jms/AlertQueue
    - Bus name: Select "SAS Messaging Bus" from the drop-down menu.
    - Queue name: Select "AlertQueue" from the drop-down menu. (The name of the new destination created above).
    - Time to live: not-set (If "not-set", leave the field empty; otherwise enter time in milliseconds).
    - Priority: not-set (If "not-set", leave the field empty; otherwise enter priority in range 0-9).
    - **Leave all other fields on the page empty or with their default pre-filled values and click Ok.**
11. Be sure to save your work and perform a Full Resynchronize to ensure the new definitions are propagated from dmgr to nodeagent.

Create a JMS Queue "WorkflowQueue" and Service Integration Bus Destination "WorkflowQueue" at the "CELL" level.

Create a new JMS Queue "WorkflowQueue" and Service Integration Bus Destination "WorkflowQueue" at scope "HOSTNAMECell01" on SIBus "SAS Messaging Bus".

1. Create the Service Integration Bus Queue destination "WorkflowQueue", if it doesn't already exist.
2. Navigate to Service integration->Buses->, click the "SAS Messaging Bus" link, then click "Destinations".
3. Click New and ensure the Queue radio button is selected.
4. Enter "WorkflowQueue" (without the "quotes") in the required "Identifier" field.
5. On the Assign the queue to a bus member page, select the entry for "Node=HOSTNAMENode01:Server=server1" from the Bus member drop-down menu.
6. Next, create the JMS Queue with a Name of "WorkflowQueue" to be associated with the destination just created.
7. Navigate to "Resources->JMS->Queues" menu on the WAS Admin Console.
8. On the Queues page, be sure the scope setting "Cell=HOSTNAMECell01" is selected from the drop-down menu and click New.
9. Make sure the "Default messaging provider" radio button is selected and click Ok.
10. For the input fields on the Queues->Default messaging provider->New configuration page, complete the following fields:
    - Name: WorkflowQueue
    - JNDI Name: sas/jms/WorkflowQueue
    - Bus name: Select "SAS Messaging Bus" from the drop-down menu.
    - Queue name: Select "WorkflowQueue" from the drop-down menu. (The name of the new destination created above).
    - Time to live: not-set (If "not-set", leave the field empty; otherwise enter time in milliseconds).
    - Priority: not-set (If "not-set", leave the field empty; otherwise enter priority in range 0-9).
    - **Leave all other fields on the page empty or with their default pre-filled values and click Ok.**
11. Be sure to save your work and perform a Full Resynchronize to ensure the new definitions are propagated from dmgr to nodeagent.

Create a JMS Queue "WorkflowCommandQueue" and Service Integration Bus Destination "WorkflowCommandQueue" at the "CELL" level.

Create a new JMS Queue "WorkflowCommandQueue" and Service Integration Bus Destination "WorkflowCommandQueue" at scope "HOSTNAMECell01" on SIBus "SAS Messaging Bus".

1. Create the Service Integration Bus Queue destination "WorkflowCommandQueue", if it doesn't already exist.
2. Navigate to Service integration->Buses->, click the "SAS Messaging Bus" link, then click "Destinations".
3. Click New and ensure the Queue radio button is selected.
4. Enter "WorkflowCommandQueue" (without the "quotes") in the required "Identifier" field.
5. On the Assign the queue to a bus member page, select the entry for "Node=HOSTNAMENode01:Server=server1" from the Bus member drop-down menu.
6. Next, create the JMS Queue with a Name of "WorkflowCommandQueue" to be associated with the destination just created.
7. Navigate to "Resources->JMS->Queues" menu on the WAS Admin Console.
8. On the Queues page, be sure the scope setting "Cell=HOSTNAMECell01" is selected from the drop-down menu and click New.
9. Make sure the "Default messaging provider" radio button is selected and click Ok.
10. For the input fields on the Queues->Default messaging provider->New configuration page, complete the following fields:
    - Name: WorkflowCommandQueue
    - JNDI Name: sas/jms/WorkflowCommandQueue
    - Bus name: Select "SAS Messaging Bus" from the drop-down menu.
    - Queue name: Select "WorkflowCommandQueue" from the drop-down menu. (The name of the new destination created above).
    - Time to live: not-set (If "not-set", leave the field empty; otherwise enter time in milliseconds).
    - Priority: not-set (If "not-set", leave the field empty; otherwise enter priority in range 0-9).
    - **Leave all other fields on the page empty or with their default pre-filled values and click Ok.**
11. Be sure to save your work and perform a Full Resynchronize to ensure the new definitions are propagated from dmgr to nodeagent.

Create a JMS Queue "WorkflowEventsQueue" and Service Integration Bus Destination "WorkflowEventsQueue" at the "SERVER" level.

Create a new JMS Queue "WorkflowEventsQueue" and Service Integration Bus Destination "WorkflowEventsQueue" at scope "HOSTNAMECell01.HOSTNAMENode01.server1" on SIBus "SAS Messaging Bus".

1. Create the Service Integration Bus Queue destination "WorkflowEventsQueue", if it doesn't already exist.
2. Navigate to Service integration->Buses->, click the "SAS Messaging Bus" link, then click "Destinations".
3. Click New and ensure the Queue radio button is selected.
4. Enter "WorkflowEventsQueue" (without the "quotes") in the required "Identifier" field.
5. On the Assign the queue to a bus member page, select the entry for "Node=HOSTNAMENode01:Server=server1" from the Bus member drop-down menu.
6. Next, create the JMS Queue with a Name of "WorkflowEventsQueue" to be associated with the destination just created.
7. Navigate to "Resources->JMS->Queues" menu on the WAS Admin Console.
8. On the Queues page, be sure the scope setting "Node=HOSTNAMENode01, Server=server1" is selected from the drop-down menu and click New.
9. Make sure the "Default messaging provider" radio button is selected and click Ok.
10. For the input fields on the Queues->Default messaging provider->New configuration page, complete the following fields:
    - Name: WorkflowEventsQueue
    - JNDI Name: sas/jms/app/WorkflowEventsQueue
    - Bus name: Select "SAS Messaging Bus" from the drop-down menu.
    - Queue name: Select "WorkflowEventsQueue" from the drop-down menu. (The name of the new destination created above).
    - Time to live: not-set (If "not-set", leave the field empty; otherwise enter time in milliseconds).
    - Priority: not-set (If "not-set", leave the field empty; otherwise enter priority in range 0-9).

- **Leave all other fields on the page empty or with their default pre-filled values and click Ok.**

11. Be sure to save your work and perform a Full Resynchronize to ensure the new definitions are propagated from dmgr to nodeagent.

Create Data Source "SharedServices" at the "SERVER" level.

Use the "Resources->JDBC" menu on the WAS Admin Console to create the "SharedServices" JDBC Data Source definition.
NOTE: surrounding "quotes" for any of the following settings should **NOT** be entered into the WAS Admin input fields.

1. Before creating the JDBC provider SharedServices SAS JDBC Provider and JDBC Data source SharedServices, you will need to create a "JAAS - J2C authentication data Alias" entry named "SASJAASAlias". This will contain the User ID and password used to authenticate the connection to the "SharedServices" data source. This alias will be referenced by during creation of the new JDBC data source "SharedServices".
2. Navigate to the Security->Secure administration, applications, and infrastructure->Java Authentication and Authorization Service->J2C authentication data link.
3. Create a new JAAS - J2C Alias entry for "SASJAASAlias" (without the "quotes").
4. On the JAAS - J2C authentication data New page, enter the information below into the required fields:
   - Alias: SASJAASAlias
   - User ID: <Enter the user ID for connection to the database (Use the SAS trusted user ID when datasource is for SAS TableServer)>
   - Password: <Enter the password for connection to the database (Use the **unencoded-password** for the SAS trusted user ID when datasource is for SAS TableServer)>.
     NOTE: WAS will store this password using it's own encoding mechanism.
5. Next, navigate to Resources->JDBC->JDBC Providers link.
6. Ensure scope is set to "SERVER" level scope (HOSTNAMECell01.HOSTNAMENode01.server1) before creating the new "SharedServices SAS JDBC Provider" JDBC Provider.
7. Enter the following values into the required fields:
   - Database type: Choose "User-defined" from the drop-down menu
   - Implementation class name: "com.sas.tkts.TKTSConnectionPoolDataSource" (without the "quotes")
   - Name: "SharedServices SAS JDBC Provider" (without the "quotes")
8. On the "Enter database class path information" page, enter each of the following classpath entries as a separate, new-line delimited string into the Classpath input box:
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\icu4j.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\log4j.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.core.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.core.nls.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.icons.contents.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.icons.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.icons.nls.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.intrnet.javatools.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.intrnet.javatools.nls.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.nls.collator.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.oda.tkts.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.oda.tkts.nls.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.security.sspi.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.svc.connection.jar
   - C:\SAS\Config\Lev1\Web\Applications\SASSharedServices9.2\JDBCDrivers\sas.svc.connection.nls.jar
9. After reviewing the Summary information, you should be placed back on the JDBC providers page.
10. Ensure server scope is set to "Node=HOSTNAMENode01,Server=server1" and click the link for the newly created "SharedServices SAS JDBC Provider" JDBC provider.
11. From the "JDBC Providers->SharedServices SAS JDBC Provider" page, click the "Data sources" link.
12. Create a data source "SharedServices" using the following values for the required fields:
    - Data source name: "SharedServices" (without the "quotes")
    - JNDI name: "sas/jdbc/SharedServices" (without the "quotes")
    - Component-managed authentication alias and XA recovery authentication alias: SASJAASAlias (i.e. choose the JAAS - J2C authentication data alias created above in the drop-down menu)
13. Leave the "Data store helper class name" field as-is, it has been pre-filled with the WebSphere default helper class.
14. Be sure to uncheck the "Use this data source in container managed persistence (CMP) checkbox".
15. Review the Summary, you should be transferred back to the "JDBC providers->SharedServices SAS JDBC Provider->Data sources" page.
16. Click the link for the newly created "SharedServices".
17. Click the Custom Properties link. Enter a custom property name and value for the "SharedServices" data source to match each row in the following table.
    NOTE: A custom property with a name of "connectionSharing" and a value of "1" may already be present. Although this property does not appear in the table, it is optional and can safely remain. With exception of this property, ensure that only those with names and values matching rows in the table are present. If required, delete any that do not match using the WAS Admin Console.
18. After entering all custom properties and checking for accuracy, save your work and perform a Full Resynchronize to ensure the dmgr and nodeagent are updated.

Custom Properties for Data Source "SharedServices"	
Custom Property Name:	**Custom Property Value:**
user	Enter the SAS trusted user ID
password	Enter the **unencoded-password** for the SAS trusted user ID
serverName	HOSTNAME.example.com

portNumber	2171
serverUrl	jdbc:sastkts://HOSTNAME.example.com:2171
dataSourceName	SharedServices
databaseName	SharedServices
serverName	HOSTNAME.example.com
portNumber	2171
stmtPooling	0
constring	(DSN=SharedServices)

Deployment for application "SASWebInfrastructurePlatformServices9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASWebInfrastructurePlatformServices9.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.wip.services9.2.ear" to locate the EAR for the SAS "SASWebInfrastructurePlatformServices9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASWebInfrastructurePlatformServices9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASWebInfrastructurePlatformServices9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 2.
   - From the Enterprise Applications menu, select the "SASWebInfrastructurePlatformServices9.2" link.
   - Click the "Startup behavior" link.
   - Enter "2" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASWebInfrastructurePlatformApplications9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASWebInfrastructurePlatformApplications9.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.wip.apps9.2.ear" to locate the EAR for the SAS "SASWebInfrastructurePlatformApplications9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASWebInfrastructurePlatformApplications9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASWebInfrastructurePlatformApplications9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.

- You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
- You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work

8. Set the application startup order to 3.
   - From the Enterprise Applications menu, select the "SASWebInfrastructurePlatformApplications9.2" link.
   - Click the "Startup behavior" link.
   - Enter "3" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASStoredProcessApplication9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASStoredProcessApplication9.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.storedprocess9.2.ear" to locate the EAR for the SAS "SASStoredProcessApplication9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASStoredProcessApplication9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASStoredProcessApplication9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "SASStoredProcessApplication9.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASContentServer9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASContentServer9.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.wip.scs9.2.ear" to locate the EAR for the SAS "SASContentServer9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASContentServer9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASContentServer9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work

8. Set the application startup order to 4.
   - From the Enterprise Applications menu, select the "SASContentServer9.2" link.
   - Click the "Startup behavior" link.
   - Enter "4" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASThemes9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASThemes9.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.themes.ear" to locate the EAR for the SAS "SASThemes9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASThemes9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASThemes9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 1.
   - From the Enterprise Applications menu, select the "SASThemes9.2" link.
   - Click the "Startup behavior" link.
   - Enter "1" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASPortal4.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASPortal4.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.portal4.2.ear" to locate the EAR for the SAS "SASPortal4.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASPortal4.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASPortal4.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 5.
   - From the Enterprise Applications menu, select the "SASPortal4.2" link.
   - Click the "Startup behavior" link.
   - Enter "5" (without the "quotes") in the required "Startup order" field.

- Allow all other fields should remain empty or with their pre-filled values.

9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASPackageViewer4.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASPackageViewer4.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.packageviewer4.2.ear" to locate the EAR for the SAS "SASPackageViewer4.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASPackageViewer4.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASPackageViewer4.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "SASPackageViewer4.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASWebReportStudio4.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASWebReportStudio4.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging/sas.webreportstudio4.2.ear" to locate the EAR for the SAS "SASWebReportStudio4.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASWebReportStudio4.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASWebReportStudio4.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "SASWebReportStudio4.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASSharedServices9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASSharedServices9.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.shared9.2.ear" to locate the EAR for the SAS "SASSharedServices9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASSharedServices9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASSharedServices9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "SASSharedServices9.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "HOSTNAMEDashboard4.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "HOSTNAMEDashboard4.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging/sas.bidashboard4.2.ear" to locate the EAR for the SAS "HOSTNAMEDashboard4.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "HOSTNAMEDashboard4.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "HOSTNAMEDashboard4.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "HOSTNAMEDashboard4.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASWebDoc9.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASWebDoc9.2" SAS web application. Perform the following steps to deploy this

application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.webdocmd9.2.ear" to locate the EAR for the SAS "SASWebDoc9.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASWebDoc9.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASWebDoc9.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "SASWebDoc9.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Deployment for application "SASWebOLAPViewer4.2" on "server1".

Because you have NOT requested auto-configuration of WAS, or have NOT requested auto-deployment of the SAS web applications, you must use the WAS Admin Console to manually deploy the "SASWebOLAPViewer4.2" SAS web application. Perform the following steps to deploy this application and make it ready for SAS operations:

1. Navigate to Enterprise Applications->Install and under the Local file system radio button, enter "C:\SAS\Config\Lev1\Web\Staging\sas.webolapviewer4.2.ear" to locate the EAR for the SAS "SASWebOLAPViewer4.2" application.
2. Allow all other fields on this page to remain with the pre-filled values and click Next.
3. On the "Select installation options" page, allow all fields to remain with pre-filled defaults, with exception of "Deploy Web services" checkbox. Check this box to ensure the Web services deploy tool wsdeploy runs during application installation.
4. Enter "SASWebOLAPViewer4.2" (without the "quotes") into the Application name field.
5. On the "Map modules to servers page", make sure the fully-qualified target server has been selected from the drop-down menu. It should be named "WebSphere:cell=HOSTNAMECell01,node=HOSTNAMENode01,server=server1".
6. Examine the summary page for accuracy, click Finish, and save your work after WAS indicates the application has been successfully deployed.
7. Update the Classloader policy and mode for the EAR and the Classloader mode for **EACH** WAR module contained in the EAR.
   - From the Enterprise Applications menu, select the "SASWebOLAPViewer4.2" link.
   - Click the "Classloading and update detection" link for the application EAR and ensure the "Classes loaded with application class loader first" radio button is selected for this EAR.
     NOTE: By default, the WAR Classloading policy at the EAR level is set to "MULTIPLE", so no action is required.
   - You may be forced to enter a value in the "Polling intervalfor updated files", if so please enter "3" (without the "quotes"), since 3 seconds is the WAS default.
   - Back on the Enterprise Applications page for this application, click the "Manage Modules" link.
   - You now must select **EACH** WAR module, choosing "Classes loaded with application class loader first" (i.e. "PARENT_LAST") from the drop-down menu for "Class loader order".
   - You may wait until all WAR module Classloader modes have been set to "PARENT_LAST" before saving your work
8. Set the application startup order to 100.
   - From the Enterprise Applications menu, select the "SASWebOLAPViewer4.2" link.
   - Click the "Startup behavior" link.
   - Enter "100" (without the "quotes") in the required "Startup order" field.
   - Allow all other fields should remain empty or with their pre-filled values.
9. You should perform a Full Resynchronize to ensure the new application definition is propagated between dmgr and the nodeagent.
10. **NOTE:** Wait until all SAS web applications have been successfully deployed, then use the WAS Admin Console to start them by selecting all applications and clicking Start.

Starting the WebSphere Server(s)

- **Starting the Dmgr server:** Issue "C:\Program Files\IBM\WebSphere\AppServer\bin\startManager.bat" -profileName "Dmgr01" or, if on Windows, by starting the the Windows service.
- **Starting the Node Agent server:** Issue "C:\Program Files\IBM\WebSphere\AppServer\bin\startNode.bat" -profileName "AppSrv01" or, if on Windows, by starting the Windows service.

- **Starting the Application Server:** Issue "C:\Program Files\IBM\WebSphere\AppServer\bin\startServer.bat" "server1" -profileName "AppSrv01"

SAS Web Applications

SAS Themes

Validation steps	SAS Themes are used by SAS theme-enabled web applications. If you go to the logon screen for the Portal or Web Report Studio and see images then SAS Themes are working properly.

SAS Web Infrastructure Platform

Log directory	C:\SAS\Config\Lev1\Web\Logs
Validation steps	The SAS Web Infrastructure Platform is used by other SAS web applications, so if the validation steps for SAS web applications are successful, then the SAS Web Infrastructure Platform is working properly.

SAS Content Server

Log file	C:\SAS\Config\Lev1\Web\Logs\SASContentServer.log
Validation steps	1. In the SAS Management Console, on the Plug-ins tab, select "+" to expand the Server Manager node. 2. Highlight the SAS Content Server. 3. Right mouse click this server and select Validate. 4. Log in using valid credentials for an unrestricted user defined in the SAS Metadata Server. 5. You should see a Validation Successful message.

SAS Stored Process Web Application

URL	http://HOSTNAME.example.com:9080/SASStoredProcess/do
Log file	C:\SAS\Config\Lev1\Web\Logs\SASStoredProcess9.2.log
Validation steps	1. Access the URL provided above. 2. Log in using the HOSTNAME\sasdemo credentials provided during configuration. 3. On the Welcome page, click the "Sample: Shoe Sales by Region" link to access the custom input form for this sample stored process. 4. Select an ODS style or accept the default value, and click the "Display SAS Output" button. 5. The output from the stored process is a static table and a clickable chart. Click a bar segment on the chart to drill down to the data for that country.

SAS Web Report Studio

URL	http://HOSTNAME.example.com:9080/SASWebReportStudio
Log directory	C:\SAS\Config\Lev1\Web\Logs\SASWebReportStudio4.2.log For more details about how to modify your log format and to see what logging entails, see "Logging for SAS Web Applications" in the *SAS Intelligence Platform: Web Application Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. Access the URL provided above. 2. Login using valid credentials defined in the SAS Metadata Server.

SAS Web OLAP Viewer for Java

;

URL	http://HOSTNAME.example.com:9080/SASWebOLAPViewer
Log file	C:\SAS\Config\Lev1\Web\Logs\SASWebOLAPViewer4.2.log For more details about how to modify your log format and to see what logging entails, see "Logging for SAS Web Applications" in the *SAS Intelligence Platform: Web Application Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. Access the URL provided above. 2. Login using valid credentials defined in the SAS Metadata Server.

SAS Information Delivery Portal

URL	http://HOSTNAME.example.com:9080/SASPortal
Log file	C:\SAS\Config\Lev1\Web\Logs\SASPortal4.2.log For more details about how to modify your log format and to see what logging entails, see "Logging for SAS Web Applications" in the *SAS Intelligence Platform: Web Application Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. Access the URL provided above. 2. Login using the HOSTNAME\sasdemo credentials provided during configuration. 3. Since the Portal samples were installed, verify that a Home page was created and contains two portlets, My Collections and Bookmarks. 4. Perform a search to verify that you have access to SAS content.

Notes:

- The SAS Portlets are located at: C:\SAS\Config\Lev1\Web\Applications\SASPortlets4.2\Deployed
- If your network has a proxy server, you must specify the following parameters on the Application Server start-up command so that **syndication channels** and the **URL Display Portlet** can access content outside of your network.
  - -Dhttp.proxyHost=*host*
  - -Dhttp.proxyPort=*port*

SAS Shared Services

Log directory	C:\SAS\Config\Lev1\Web\Logs
Validation steps	1. Access the Comment Service. 2. Log in using the HOSTNAME\sasdemo credentials provided during configuration. 3. Verify that you can add a comment.

Notes:

SAS BI Dashboard 4.2

URL	http://HOSTNAME.example.com:9080/HOSTNAMEDashboard
Log directory	C:\SAS\Config\Lev1\Web\Logs\HOSTNAMEDashboard4.2.log For more details about how to modify your log format and to see what logging entails, see "Logging for SAS Web Applications" in the *SAS Intelligence Platform: Web Application Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
Validation steps	1. Grant the user SAS Demo User access to SAS BI Dashboard 4.2. In SAS Management Console, on the Plug-ins tab, click on the User Manager node. 2. Right click on "BI Dashboard Administrators" and select "Properties." 3. Click on the "members" tab 4. Double click on SAS Demo User to add it to the group. 5. Click OK. 6. Access the URL provided above. 7. Login using the HOSTNAME\sasdemo credentials provided during configuration. 8. Create a new indicator using the New Indicator option. 9. Add this indicator to a new or existing dashboard and verify that you have access to this dashboard by adding the BI Dashboard Portlet to a Portal page.

Notes:

- The SAS BI Dashboard 4.2 sample files are located at: C:\Program Files\SAS\HOSTNAMEDashboard\4.2\SampleData. Please refer the sample files install guide for more information.

SAS Help Viewer Metadata Configuration

URL	http://HOSTNAME.example.com:9080/SASWebDoc
Validation steps	Use the URL above and examine the Installed Products Page, if there were any errors configuring help content, they will be reported on that page.

Web Application Custom Content

If you wish to add custom content into SAS web applications, you may do so by placing your custom content into the appropriate custom content directory structure for the given application, and then running the SAS Deployment Manager to rebuild web applications. The custom content root directory for a given web application is

```
C:\SAS\Config\Lev1\Web\Common\Server Name\Application Name\CustomContent
```

For example,

```
C:\SAS\Config\Lev1\Web\Common\SASServer1\SASWebReportStudio4.2\CustomContent
```

Within that directory, there are subdirectories (for example, `ears` or `wars`) corresponding to specific archive types (ear and war files, respectively). Under the archive type directories, there are subdirectories for each specific archive—these are the root directories for each archive within the application. Custom content should be placed in the archive's directory tree corresponding to where the content should appear within the archive.

For example, to add the `ear_addon.xml` file into the `addons` directory in the `sas.webreportstudio` ear file, create the

```
...\CustomContent\ears\sas.webreportstudio\addons
```

directory, and place `ear_addon.xml` in it. Our process knows which war files are contained within ear files, so if you want to add `war_addon.jar` into the `WEB-INF/lib` directory in the `sas.webreportstudio` war file within the `sas.webreportstudio` ear file, then create

```
...\CustomContent\wars\sas.webreportstudio\WEB-INF\lib
```

directory, and place `war_addon.jar` in it.

Once all your custom content is ready to be consumed, run the SAS Deployment Manager and choose to rebuild web applications. Doing so will rebuild the web applications, inserting the custom content into the archives under the appropriate paths. Note that if custom content has the same path and name of content normally included in the archive, then the custom content takes precedence.

SAS Metadata Bridges

The SAS Metadata Import or Export function of SAS Management Console and SAS Data Integration Studio will prompt for the location of a SAS license file. Therefore, a valid SAS license file must be accessible from the client machine. The license file may be available in the sid_files directory at the root of your SAS Software Depot or it may be included with your SAS Software Order E-mail. If it is with your SAS Software Order E-mail, then you should follow the instructions included in that e-mail for where and how to store your SAS license file.

Completing Your Deployment

In order to complete your deployment on this machine, certain manual steps have to be performed. The SAS Content Server must be running before you attempt these steps.

Loading Content to the SAS Content Server

- Since you selected to not automatically deploy your SAS Web Applications, manual steps are required to ensure the proper content is loaded to the SAS Content Server. The file, C:\SAS\Config\Lev1\Web\Utilities\manualLoadContent.bat, has been created to facilitate this process.
- Click here to run this script to automatically load content for the SAS Web Applications that require it.

The steps detailed above resulted in the creation of log files, which can be found in C:\SAS\Config\Lev1\Logs\Configure. Review the contents of the scripts you executed for the name of the log files that were created. Review these logs and make sure there were no errors. Each execution of these scripts results in the logs from previous executions being overwritten. Therefore, prior to rerunning these scripts, you may wish to make appropriate backups of the logs.

These manual steps required the use of the SAS Administrator account. The scripts contain the SAS Administrator's credentials with the password encrypted at the sas002 level. The scripts are not intended for further use once your deployment is complete. Since these scripts contain the SAS Administrator's credentials, it is highly recommended that you delete them once your deployment is complete.

Obtaining Additional Information

Additional SAS Intelligence Platform documentation is available from the SAS 9.2 administration documentation web site.

There may be additional configuration steps required for your SAS Foundation software. Consult the SAS Foundation Configuration instructions available for your operating system for product-specific post-installation steps. The configuration instructions are available from the SAS Install Center web site.

Backing up your SAS Metadata Server

After completing your SAS Intelligence Platform configuration, we strongly recommend that you create a backup of your newly configured SAS Metadata Server. If you have adequate disk space to accommodate a copy of all of your metadata repositories on the metadata server host, use the MetadataServer.bat script provided in the C:\SAS\Config\Lev1\SASMeta\MetadataServer directory. To execute the script, type:

MetadataServer.bat backup

The script copies all critical metadata server files to a SASBackup subdirectory of the MetadataServer directory.

If disk space is an issue, then use the SAS Backup wizard provided in the SAS Management Console Metadata Manager or the %OMABAKUP macro to perform the backup. Both of these tools copy all critical metadata server files and permit you to specify a different backup destination. For more information, see "About the Backup Wizard" and "About %OMABAKUP" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.

Regular metadata server backups using SAS backup tools are crucial to ensuring the health of your SAS 9.2 Intelligence Platform configuration, but they are just one part of backing up a SAS Intelligence Platform configuration. To devise a backup strategy for your entire SAS Intelligence Platform configuration, see "Best Practices for Backing Up and Restoring Your System" in the *SAS Intelligence Platform: System Administration Guide* at http://support.sas.com/documentation/onlinedoc/intellplatform/index.html#intell92.
