



IBM Software Group

Understanding and troubleshooting common issues of Synchronization service in WebSphere Application Server V7.x/8.x

Vikram Thommandru (vikramt@us.ibm.com)

Phuong Ngo (pngo@us.ibm.com)

WebSphere Application Server L2 Support

08/28/2013



WebSphere® Support Technical Exchange



Agenda

- *Overview of synchronization service*
- *Possible ways to synchronize repositories*
- *Types of synchronization operations*
- *Reviewing the trace.*
- *Trouble shooting and debugging of synchronization service*
- *Questions and answer.*

What is synchronization service :

- The synchronization service ensures that a file set on each node matches that on the deployment manager node. This service promotes consistent configuration data across a cell.
 - ▶ This service runs in the deployment manager and node agents.
 - ▶ It ensures that configuration changes made to the cell repository are propagated to the appropriate node repositories.
 - ▶ The cell repository is the master repository.
 - ▶ Configuration changes made to node repositories are not propagated up to the cell.
 - ▶ The default behavior, which is enabled, is for each node agent to periodically run a synchronization operation.



Possible ways to synchronize repositories :

- ▶ Auto sync (performed by node agent in the background periodically)
- ▶ User explicitly initiated synchronization (through console and wsadmin, or MBean API, while node agent is running)
- ▶ syncNode.bat/sh (require node agent to be stopped)
- ▶ Synchronization before Application Server startup (if Startup Synchronization flag is checked)
- ▶ Synchronization during node federation (performed by addNode process as one of the tasks during node federation)
- ▶ During restart JMX call on NodeAgent MBean, if the first parameter, syncFirst, is set to true
- ▶ Normal (partial) synchronization (only synchronization files dmgr thinks have changed)
- ▶ Full synchronization (synchronize all files in the repositories)

How does synchronization actually work

- Both the node agent and deployment manager keep a calculated digest for each file in the configuration they manage (master cell repository managed by deployment manager, local node repository managed by node agent). These digest values are stored in memory.
- Also stored in memory are "epochs" for the folders in the repository and for the repository overall. These epochs are used to determine whether any files in the directory have changed.
- If a change to a configuration file is made through the administration programs (Administration console, wsadmin, or other), then the overall repository epoch and the epoch for the folder in which that file resides is modified. During configuration synchronization operations, if the repository epoch has changed since the previous synchronization operation, then individual folder epochs will be compared. If the epochs for corresponding node and cell directories do not match, then the digests for all files in the directory are recalculated, including that changed file. If a change to a configuration file is made by editing the file and bypassing the administration programs, then the digest is not recalculated because the epochs for the directories continue to match.
- The algorithm for synchronization by default only compares these epochs between node agent and deployment manager. The files are not read on every synchronization operation. So, by DEFAULT, manual edits to the configuration files in the cell repository are not picked up during "normal synchronization" operations. This was a compromise reached to allow best performance under normal conditions since editing the files manually is not a recommended best practice.



More details on how synchronization works

- Synchronization operations are invoked by node agent in the cases of auto sync or user explicitly initiated synchronization, etc; or by the syncNode/addNode processes when syncNode.bat/sh or addNode.bat/sh is invoked.
- In all synchronization scenarios, the process initializing the synchronization operation communicates with the dmgr. The dmgr retrieves information about the state of master repository and compare that with the node repository. A list of changed folders is returned back to the node agent, after comparing the epochs of the two repositories. Next, for each folder in the list, dmgr compares the digests of the documents to see if files are indeed different. The changed files are transferred to the node via file transfer and checked into node repository by the node agent.
- A synchronization operation can be started if there is not another synchronization operation already going on; otherwise you wait for the on-going synchronization operation to finish before a new one is started.

Detailed steps:

1. Initialize:

- Reset folder epoch values if necessary.
- Send out websphere.nodesync.initiated notification.
- Spawn a new thread to do the actual synchronization work.

2. Synchronize:

- Synchronization is an iterative process (max number of iteration is determined by SYNC_ITERATION_LIMIT, which is hard coded to 3).
- If the repository epoch for the dmgr and node agent are the same, then synchronization is done, go to step 3.
- If the repository epochs do not match, then something has changed and you have some work to do:
 - a) First you need to get a list of changed folders in master repository:
 - This is achieved by making the getModifiedFolders() JMX call on CellSync MBean, which exist in dmgr process.
 - The epochs for the folders in node repository are passed to dmgr during this call.
 - dmgr compares the epochs from the node repository with those in its own repository to decide which folders have changed.
 - dmgr returns a list of modified folders back, as the result of getModifiedFolders() call, along with the type of changes occurred to these folders (deleted, modified, created)

Continued..

b) For the list of changed folders:

- 1) If the folder is deleted in master repository, delete it from node repository
- 2) If the folder is modified/created in master repository, then you need to compare the digests:
 - Get the digests from the local (node) repository.
 - Invoke getFolderSyncUpdates JMX call on CellSync MBean, passing the digests for dmgr to compare with.
 - The dmgr compares the digests to determine which documents have been updated.
 - As the result of invoke getFolderSyncUpdates call, a list of files changed (created, modified, deleted) is returned.
 - Node agent downloads changed documents by way of file transfer.

c) Go through the list of changed documents and check each document into node repository:

- *After 2.c, you go back and check repository epochs. If they match, then you are done. Repository has been synchronized. Otherwise, master repository has changed during this synchronization operation, you need to repeat 2.a to 2.c to synchronize up new changes.*
- *Note, you only repeat synchronization for a maximum of 3 times. After that, you stop and output a message:*

ADMS0023I: A synchronization operation reached the iteration limit

3. Post synchronization operations:

- Send synchronization completion notification.
- Depending on which files are updated in the node repository, application management code might be invoked to expand the binary EAR to installedApps.

There are two types of synchronization:

- *Normal Synchronization propagates only the files modified using an administrative tool (WebSphere Application Server Administrative Console or wsadmin) from the Network Deployment manager to the appserver node.*
 - ▶ **Ways to perform normal synchronization:**
 1. Auto sync.
 2. Selecting "Synchronize" button in the console, or if invoke "sync" operation on NodeSync Mbean.
 3. Synchronization before Application Server startup (If this is neither the first synchronization operation since node agent starts up and nor the first synchronization operation since a "refreshRepositoryEpoch" JMX call is invoked.
- *Full Resynchronization propagates all files from the Network Deployment Manager to the AppServer node that differ in timestamps.*
 - ▶ **Ways to perform full synchronization:**
 1. Select "Full Resynchronize" button in the console
 2. Invoke "refreshRepositoryEpoch" operation on ConfigRepository MBean in node agent process, before invoking "sync" operation on the Node Sync Mbean
 3. Running syncNode.sh/bat from Application Server node.
 4. Synchronization before Application Server startup (If this is the very first synchronization operation node agent performs since node agent starts up, or If this is the very first synchronization operation node agent performs since a "refreshRepositoryEpoch" JMX call)

Console navigation to sync buttons and file sync service in the console :

Integrated Solutions Console Welcome - Testing Help | Logout

Cell=vikramCell01, Profile=Dmgr01 Close page

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
- Services
- Resources
- Security
- Environment
- System administration
 - Cell
 - Save changes
 - Deployment manager
 - Nodes**
 - Node agents

Nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.

Preferences

Select	Name	Host Name	Version	Discovery Protocol	Status
	vikramCellManager01	localhost	ND 7.0.0.21	TCP	
<input type="checkbox"/>	vikramNode01	localhost	ND 7.0.0.21	TCP	

Integrated Solutions Console Welcome - Testing Help | Logout

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
- Services
- Resources
- Security
- Environment
- System administration
 - Cell
 - Save changes
 - Deployment manager
 - Nodes
 - Node agents
 - Node groups
 - Centralized Instance
 - Console Preferences

Node agents > nodeagent > File synchronization service

Use this page to configure the file synchronization service. The file synchronization service runs in the deployment manager and node agent. It ensures that configuration changes made to the cell repository are propagated to the appropriate node repositories.

Configuration

General Properties

Enable service at server startup

* Synchronization interval: minutes

Automatic synchronization

Startup synchronization

Exclusions
cells/vikramCell01/namestore.xml

Additional Properties
[Custom properties](#)

What is file synchronization service exclusion option:

- ▶ Specifies files or patterns that should not be part of the synchronization of configuration data. Files in this list are not copied from the master configuration repository to the node, and are not deleted from the repository at the node.
- ▶ The default is to have no files specified.

To specify a file, use a complete name or a name with a leading or trailing asterisk (*) for a wildcard. For example:

<code>cells/cell name/nodes/node name/file name</code>	Excludes this specific file
<code>*/file name</code>	Excludes files named <i>file name</i> in any context
<code>dirname/*</code>	Excludes the subtree under <i>dirname</i>

- ▶ Press Enter at the end of each entry. Each file name appears on a separate line.
- ▶ Since these strings represent logical document locations and not actual file paths, only forward slashes are needed no matter the platform.
- ▶ Changes to the exclusion list are picked up when the node agent is restarted.

- Synchronization process always propagates files from deployment manager cell to the appserver node.
- It can't propagate files from appserver node config folder to deployment manager master repository.
- It is not recommended or best practice manually editing files inside config folder (or) dropping/removing files inside config folder.

Reviewing the trace :

- Reviewing the following trace, and then comparing against the trace gathered, greatly assists in pinpointing the problem area.
 - ▶ The following trace illustrates a synchronization operation from start to finish. The following trace level is set for both dmgr and node agent: Admin=all=enabled:Sync=all=enabled
 - ▶ All traces are taken from the node agent trace file, except the ones starting with "(dmgr trace)".

Step 1. Initialization sync:

```
[6/22/13 21:27:04:686 EDT] 2e4c94c6 NodeSyncTask > initiate: false
```

```
[6/22/13 21:27:04:686 EDT] 2e4c94c6 SyncResult > initialize
```

```
[6/22/13 21:27:04:686 EDT] 2e4c94c6 SyncResult < initialize
```

```
[6/22/13 21:27:04:686 EDT] 2e4c94c6 NodeSyncTask d setState: 1 <-- sync state: IDLE=0
```

```
COMPARE_FOLDERS=1 PROCESSING_UPDATE=2 PROCESSING_UPDATE_WITH_ERROR=3  
VERIFY_SYNC_RESULT=4
```

```
[6/22/13 21:27:04:686 EDT] 2e4c94c6 RuntimeCollab > sendNotification <-- send sync initiated  
notification
```

```
websphere.nodesync.initiated[source=WebSphere:platform=common,cell=TestNetwork,version=  
7.0,name=nodeSync,mbeanIdentifier=nodeSync,type=NodeSync,node=Test2,process=nodeagent]
```

- **Step 2. Do sync:**

```
[6/22/13 21:27:53:446 EDT] 2b3994ce NodeSyncTask d run <-- a new thread is spawn (notice the thread ID 2b3994ce
is different from the sync init thread, 2e4c94c6)
[6/22/13 21:27:53:446 EDT] 2b3994ce NodeSyncTask > doSync <-- start doSync method, which does most of the work
[6/22/13 21:27:53:446 EDT] 2b3994ce NodeSync > getCellRepositoryEpoch <-- compare dmgr's repository epoch
with node's repository epoch
```

```
(dmgr trace) [6/22/13 21:27:53:517 EDT] 55afd4cd SOAPConnectio > handleRequest <-- dmgr receives the JMX call
```

.....

```
(dmgr trace) [6/22/13 21:27:53:567 EDT] 55afd4cd SOAPConnector d Parameter: getRepositoryEpoch(type: class
java.lang.String)
```

.....

```
(dmgr trace) [6/22/13 21:27:53:577 EDT] 55afd4cd SOAPConnectio < handleRequest <-- dmgr finishes processing the
JMX call
```

- **Step 2.1 Make getModifiedFolders() JMX call:**

```
[6/22/13 21:27:53:587 EDT] 2b3994ce NodeSync > invokeGetModifiedFolders <-- method entry
```

```
(dmgr trace) [6/22/13 21:27:53:627 EDT] 478594c2 SOAPConnectio > handleRequest <-- dmgr receives the JMX call
```

.....

```
(dmgr trace) [6/22/13 21:27:53:647 EDT] 478594c2 SOAPConnector d Parameter: getModifiedFolders(type: class
java.lang.String)
```

.....

```
(dmgr trace) [6/22/13 21:27:54:878 EDT] 478594c2 SOAPConnectio < handleRequest <-- dmgr finishes processing the
JMX call
```

.....

```
[6/22/13 21:27:54:878 EDT] 2b3994ce NodeSync d modifiedFolders <-- this is a list of folders changed in master
repository in dmgr
```

cells/TestNetwork/applications/perfServletApp.ear added
 cells/TestNetwork/applications/perfServletApp.ear/deployments added
 cells/TestNetwork/applications/perfServletApp.ear/deployments/perfServletApp added
 cells/TestNetwork/applications/perfServletApp.ear/deployments/perfServletApp/META-INF added

.....

[6/22/13 21:27:54:878 EDT] 2b3994ce NodeSync < invokeGetModifiedFolders <-- *method exit*

Step 2.2.2 Make getFolderSyncUpdate() JMX call:

[6/22/13 21:27:54:899 EDT] 2b3994ce NodeSync > invokeGetFolderSyncUpdate <-- *method entry*

.....

(dmgr trace) [6/22/13 21:27:54:919 EDT] 478094c2 SOAPConnectio > handleRequest <-- *dmgr receives the JMX call*

.....

(dmgr trace) [6/22/13 21:27:54:949 EDT] 478094c2 SOAPConnector d Parameter: getFolderSyncUpdates(type: class java.lang.String)

.....

(dmgr trace) [6/22/13 21:27:55:179 EDT] 478094c2 SOAPConnectio < handleRequest <-- *dmgr finishes processing the JMX call*

.....

[6/22/13 21:27:55:229 EDT] 2b3994ce FileTransferC > downloadFile (5) to InputStream.
 srcPath=cells/TestNetwork/applications/perfServletApp.ear/perfServletApp.ear53033.tmp,
 options=com.ibm.ws.management.filetransfer.client.FileTransferOptionsImpl@551d94c8 <-- *changed document is downloaded to node*

.....

[6/22/13 21:27:56:030 EDT] 2b3994ce NodeSync < invokeGetFolderSyncUpdate <-- *method exit, a list of changed documents is returned*

cells/TestNetwork/applications/perfServletApp.ear
 cells/TestNetwork/applications/perfServletApp.ear/perfServletApp.ear created
 cells/TestNetwork/applications/perfServletApp.ear/deployments
 cells/TestNetwork/applications/perfServletApp.ear/deployments/perfServletApp

Step 2.3 Process each document:

[6/22/13 21:27:56:030 EDT] 2b3994ce NodeSyncTask d processFolderSyncUpdate:
cells/TestNetwork/applications/perfServletApp.ear *<-- processing perfServletApp.ear folder*

.....

[6/22/13 21:27:56:190 EDT] 2b3994ce FileRepositor A ADMR0009I: Document
cells/TestNetwork/applications/perfServletApp.ear/perfServletApp.ear was created *<-- create the file in perfServletApp.ear folder*

.....

[6/22/13 21:27:56:210 EDT] 2b3994ce NodeSyncTask d processFolderSyncUpdate:
cells/TestNetwork/applications/perfServletApp.ear/deployments *<-- processing deployments folder*

.....

[6/22/13 21:27:56:230 EDT] 2b3994ce FileRepositor A ADMR0009I: Document
cells/TestNetwork/applications/perfServletApp.ear/deployments/perfServletApp/deployment.xml was created. *<-- create the file in deployments folder*

.....

Step 3 Post sync operations:

[6/22/13 21:27:56:971 EDT] 2b3994ce AppBinaryProc d xxxxxxxxxxxx pre: -1 xxxxxxxxxxxx *<--- app mgmt code gets invoked*

.....

[6/22/13 21:27:57:102 EDT] 2b3994ce AppBinaryProc < postProcess: -1 *<-- app mgmt code gets invoked (do not pay attention to the -1, unless sync fails when around here, when AppBinaryProcessor is called)*

.....

[6/22/13 21:27:57:102 EDT] 2b3514ce AppBinThread > run **<-- app mgmt spawn a new thread to expand the ear (note different threads: 2b3514ce 2b3994ce)**

[6/22/13 21:27:57:102 EDT] 2b3514ce AppBinThread d ---- Expand
cells/TestNetwork/applications/perfServletApp.ear/perfServletApp.ear to
[D:\was7\installedApps\TestNetwork\perfServletApp.ear]

.....
[6/22/13 21:27:57:102 EDT] 2b3994ce RuntimeCollab > sendNotification **<-- sending out sync complete notification**

websphere.nodesync.complete[source=WebSphere:platform=common,cell=TestNetwork,version=5.0,name=nodeSync,mbeanIdentifier=nodeSync,type=NodeSync,node=Test2,process=nodeagent]

.....
[6/22/13 21:27:57:242 EDT] 2b3994ce NodeSyncTask < doSync **<-- exiting doSync method, we are done**

Common Issues

- *Some common issues during file synchronization that we have seen are:*
 - ▶ Soap port, hostname/IP address due to connection and network issue
 - ▶ Security – certificate exchange, access denied, database issue
 - ▶ File permission – file corruption, incorrect permission
 - ▶ Transformations – in mixed cell environments, it often involves some sort of xforms corruption
 - ▶ Out of memory – not enough heapsize, disk space, and/or low ulimit

Synchronization - Soap

- *Example 1: Unable to sync due to admin_host is blocked*
 - ▶ ADFS0112E: File transfer has failed with the following message: Download retry limit exceeded for retrieving url http://<MACHINE HOSTNAME>:<ADMIN_HOST>/FileTransfer/transfer/download%<CELLS>%<CELLNAME>%2Fadmin-authz.xml1096113870665897826.tmp? compress=true&deleteOnCompletion=true.
 - ▶ E ADMR0100E: An error occurred when creating document cells/<CELLNAME>/admin-authz.xml: java.io.FileNotFoundException: File download failed: java.io.IOException: Error occurred during download from: download/cells/<CELLNAME>/admin-authz.xml1096113870665897826.tmp.
 - ▶ Exception: java.net.ConnectException: A remote host refused an attempted connect operation.
at
com.ibm.websphere.management.filetransfer.client.FileDownloadInputStream.getOutputStream(FileDownloadInputStream.java:230) at
com.ibm.websphere.management.repository.RepositoryInputStream.read(RepositoryInputStream.java:137) at java.security.DigestInputStream.read(DigestInputStream.java:107)
at java.io.FilterInputStream.read(FilterInputStream.java:101) at
com.ibm.ws.management.repository.FileDocument.copyDocument (FileDocument.java:452)

Synchronization – Soap cont.

- *The error occurred during file synchronization when it tried to transfer files from the dmgr to the node level, however, the admin_host appeared to be blocked by firewall or not available to for transfer.*

- *Suggestions:*
 - ▶ clean up wstemp, temp and config/temp for both dmgr and problematic node
 - ▶ On the node machine, do telnet test: telnet <dmgr hostname> <WC_ADMINHOST port>
 - ▶ Make sure the port is open for file transfer during syncNode operation.



Synchronization – Soap cont.

■ *Example 2: Network issue*

- ▶ SOAPConnector < reconnect Exit [SOAPException: faultCode=SOAP-ENV:Client; msg=Error opening socket: java.net.SocketException: Connection reset; targetException=java.lang.IllegalArgumentException: Error opening socket: java.net.SocketException: Connection reset] at org.apache.soap.transport.http.SOAPHTTPConnection.send(SOAPHTTPConnection.java:475) at org.apache.soap.rpc.Call.WASinvoke(Call.java:451) at com.ibm.ws.management.connector.soap.SOAPConnectorClient\$4.run(SOAPConnectorClient.java:381) at com.ibm.ws.security.util.AccessController.doPrivileged(AccessController.java:118) at com.ibm.ws.management.connector.soap.SOAPConnectorClient.reconnect(SOAPConnectorClient.java:366) at

■ *Suggestions*

- ▶ Make sure soap port is not blocked, check hostname/IP address to ensure that it resolves to correctly
- ▶ May need to take iptrace, turn on system management trace, and other low-level trace to for further investigation

Synchronization - Security

- *Example 1: Security SSL Handshake during file synchronization*

- ▶ Caused by: com.ibm.websphere.management.exception.ConnectorNotAvailableException: [SOAPException: faultCode=SOAP-ENV:Client; msg=Error opening socket: javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error: java.security.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty; targetException=java.lang.IllegalArgumentException: Error opening socket: javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error: java.security.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty] at com.ibm.ws.management.connector.soap.SOAPConnectorClient.reconnect(SOAPConnectorClient.java:409) at com.ibm.ws.management.connector.soap.SOAPConnectorClient.<init>(SOAPConnectorClient.java:222) ... 34 more Caused by: [SOAPException: faultCode=SOAP-ENV:Client; msg=Error opening socket: javax.net.ssl.SSLException: java.lang.RuntimeException:

Synchronization – Security cont.

- *Example 2: Incorrect status on the admin console after synchronization*
 - ▶ SSLHandshakeE E SSLC0008E: Unable to initialize SSL connection. Unauthorized access was denied or security settings have expired. Exception is javax.net.ssl.SSLException: Unrecognized SSL message, plaintext connection? at com.ibm.jsse2.b.a(b.java:6) at com.ibm.jsse2.qc.a(qc.java:492) at com.ibm.jsse2.qc.unwrap(qc.java:408) at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:17)
- *Suggestions:*
 - ▶ Some general suggestions to resolve security issue file synchronization include:
 - Check password, user id and permission to make sure it's correct
 - Make sure signer certificates/trust.p12 are not expired and in its locations
 - Verify database access/connections
 - Verify soap connection to/from dmgr and node is good

Synchronization – File Permission

- *Example 1: Sync failed due to file permission*

- ▶ FileDocument E ADMR0104E: The system is unable to read document cells/<CELLNAME>/cus/<APPLICATION>/cver/BASE/cu.xml:
java.io.IOException: Permission denied
 at java.io.File.checkAndCreate(File.java:1715)
 at java.io.File.createTempFile(File.java:1803)

- *Suggestions:*

- ▶ Check the file location to make sure it has correct file permission/group
- ▶ Verify the file is not corrupted
- ▶ Clean up temp, wstemp, and config/temp for the dmgr and the problematic node
- ▶ Clear out caches by issuing: clearClassCache, osgiCfgInit and clean up javasharedresources folder

Synchronization - Transformations

- *Example 1: Exception occurred after application deployment:*
 - ▶ CWWMA7005E: An unexpected exception occurred while loading resource cells/<CELLNAME>/applications/MYAPP.ear/deployments/MYAPP/deployment.xml from the repository. This error causes problems in using this resource for application synchronization logic. AppDataMgr 3 updateDepIData failed: java.lang.NullPointerException
- *Suggestions:*
 - ▶ <http://www-01.ibm.com/support/docview.wss?uid=swg1PM83565>



Synchronization – Transformations cont

- *Example 2: Exception occurred after application deployment:*
 - ▶ Caused by: java.lang.IllegalArgumentException: Not supported: <http://www.ibm.com/xmlns/prod/xltxe-j/split-limit> at org.apache.xalan.processor.TransformerFactoryImpl.setAttribute(Unknown Source) at com.ibm.ws.management.transform.DocumentTransformer.transform(DocumentTransformer.java:331)
- *Suggestions:*
 - ▶ Transformations issues during file synchronization are rare. It often involves in a mixed cell environment or a customized properties set. To debug this issue, system management trace is required as well as additional traces may be required depending on the exception.



Synchronization - discovery

- *Example 1: Unable to sync due to discovery, and/or status on the admin console are shown as 'out of sync'*
 - ▶ stack Dump = java.io.IOException: ADMD0004E: The TCP socket: 7,272 cannot be opened. At com.ibm.ws.management.discovery.transport.TcpMessenger.<init>(TcpMessenger.java:69)
- *Suggestions:*
 - ▶ clean up wstemp, temp and config/temp for both dmgr and problematic node
 - ▶ On the dmgr machine, do a telnet test: telnet <node hostname> <NODE_DISCOVERY_ADDRESS port>
 - ▶ On the node machine, do telnet test: telnet <dmgr hostname> <CELL_DISCOVERY_ADDRESS port>
 - ▶ Make sure those ports are not blocked by firewall rules, or used by other processes



General Steps to Debug

- *In general, we review dmgr SystemOut, Nodeagent SystemOut, app server SystemOut, and any ffdc logs that may give us clues on the error*
 - ▶ Basic steps to filter out cache being the issue
 - Clean up wstemp, temp, and config/temp for the dmgr and the problematic node
 - Issue: clearClassCache and osgiCfgInit
 - Clean up javasharedresources folder for windows/AIX OS
- *Most often, we look into syncNode trace:*
 - ▶ SyncNode <dmgr soap port> <dmgr hostname> -trace
- *We also review the WAS environment by reviewing the contents from the collector output*
 - ▶ temp:/WAS_home/profiles/profileName/bin/collector

Collect Data to Engage Support

- *Turn on System management trace on the deployment manager and problematic nodeagent*
 - ▶ See file synchronization mustgather for instructions to turn on trace
 - <http://www-01.ibm.com/support/docview.wss?uid=swg21140941>
 - Importance: trace needs to be turned on at *Startup*
- *In general, collector outputs for both dmgr and the problematic node.*
- *Netstat outputs if needed for connections/soap issues*

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM® Education Assistant:
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:
<http://www.ibm.com/software/support/einfo.html>

Connect with us!

1. Get notified on upcoming webcasts

Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

3. Be connected!

Connect with us on [Facebook](#)

Connect with us on [Twitter](#)



Questions and Answers