



IBM Software Group

# Top 5 Service Integration Bus Problems and their Solutions

29 October, 2014

Jhansi Kolla (jkolla@us.ibm.com)

Ty Shrake (tyshrake@us.ibm.com)



WebSphere® Support Technical Exchange



# Agenda

High level overview of Service Integration Bus

Top 5 problems in Service Integration Bus generally our customer's see in their day to day business.

Details of each problem, their causes and solutions

Summarize Highlights

Questions and Answers



# High level overview of Service Integration Bus

## Logically:

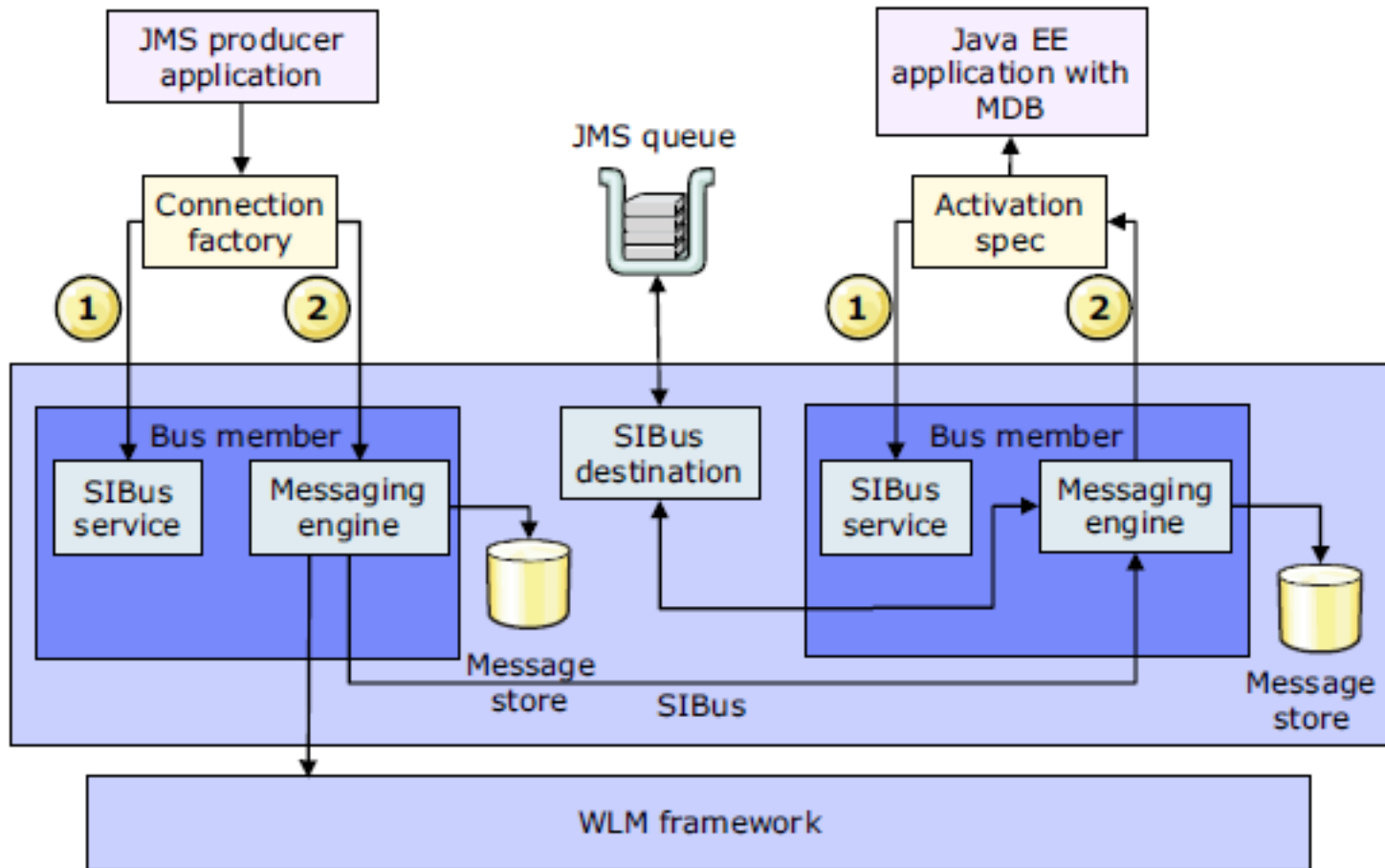
- A bus is logical entity that contains destinations (Ex: queues/topic spaces)
- Applications connect to the bus to access the destinations.
- A bus is location transparent, all destinations and their messages are available from anywhere in the bus

## Physically:

- A WAS cell contains application servers, or clusters of application servers.
- Destinations are assigned to bus members.
- Bus members run messaging engines with in application server JVM™s.
- Messaging Engines (ME) manage the runtime and persistent state of messages in the bus.
- Messaging applications form a connection to a Messaging Engine
- Applications can connect to any ME with in the bus to put/get the messages.
- Messages are routed from any ME in the bus to the application.



# High level runtime flow of SIBus Messaging:



Java™ JMS™

# Top 5 problems in SIB:

- Messaging Engine start up problems
- Application connection problems
- Message store problems
- Message flow problems
- Most common producer/consumer issues



# Messaging Engine Start Up Issues :

- It is not always the case that during application server/cluster startup the ME (Messaging Engine) is also started.
- If ME is not started ***the first thing you should do is check the messages in the SystemOut.log. If there is a failure of the ME to start the solution is probably in this presentation.***
- Check the SystemOut.log for **CWSI** type messages.

# Messaging Engine Normal Startup Sequence:

Here is a normal Messaging Engine startup sequence as it would appear in the SystemOut.log:

SibMessage I [:] CWSID0021I: Configuration reload is enabled for bus Bus1.

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSID0016I: Messaging engine wasNode01.server1-Bus1 **is in state Joined.**

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSID0016I: Messaging engine wasNode01.server1-Bus1 **is in state Starting.**

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSIS1538I: The messaging engine, ME\_UUID=FCC87D95F99CDE6E, INC\_UUID=B44E467DDE9EA5D9, **is attempting to obtain an exclusive lock on the data store.**

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSIS1543I: No previous owner was found in the messaging engines data store.

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSIS1537I: The messaging engine, ME\_UUID=FCC87D95F99CDE6E, INC\_UUID=B44E467DDE9EA5D9, **has acquired an exclusive lock on the data store.**

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSIP0212I: messaging engine wasNode01.server1-Bus1 on bus Bus1 is starting to reconcile the WCCM destination and link configuration.

SibMessage I [Bus1:wasNode01.server1-Bus1] CWSID0016I: Messaging engine wasNode01.server1-Bus1 **is in state Started.**

# Messaging Engine Startup Issues:

- Message store (file store or data store) related exceptions
- Messaging Engine starts on 2 different servers in a cluster
- Messaging Engine authentication exception
- Configuration issues



# Messaging Engine Startup Issues:

## **ME doesn't start and there are no errors related to SIB:**

Messaging Engine does not start and you do not see any errors or exceptions in SystemOut.log; not even messages indicating an attempt to start the Messaging Engine.

**Cause:** SIB service is disabled. The following single informational message is printed in the SystemOut.log files:

**CWSID0006I:** The SIB service was not enabled and will not be started.

**Solution:** Enable the SIB service using the Admin Console:

Servers -> Server Types -> WebSphere application servers -> server\_name -> [Server messaging ] SIB service

Check: "Enable service at server startup". Restart the server for the change to take effect.



# Messaging Engine Startup Issues:

## Messaging Engine will not start with CWSIS0002E and CWSIS1524E Exceptions:

Message **CWSIS0002E** is often followed by message CWSID0027, which indicates that a serious error has occurred while restarting the messaging engine, and message CWSID00016I, which indicates that the messaging engine is in a stopped state. These messages are generic in nature.

- JNDI™ name error when accessing the messaging data store.
- Invalid authentication alias information.
- Invalid database tables or tables with invalid data. For example, tables for another messaging engine data exist in data store.

SibMessage E [mename] **CWSIS0002E**: The messaging engine encountered an exception while starting. Exception: com.ibm.ws.sib.msgstore.MessageStoreRuntimeException: **CWSIS1524E**: Data source, jdbc/com.ibm.ws.sib/xxxxx, not found.

SibMessage E [mename] **CWSID0035E**: Messaging engine <mename> cannot be started; detected error reported during com.ibm.ws.sib.msgstore.impl.MessageStoreImpl start()

SibMessage I [mename] CWSID0016I: Messaging engine <mename> is in state Stopped.

**Cause:** The messaging engine cannot find the data source that it needs to access its data store.

**Solution:** The problem occurred because an incorrect value was specified in the data source JNDI name of the definition. Check the data source configuration.

# Messaging Engine Startup Issues:

## CWSIS1501E Authentication issues:

CWSIS0002E: The messaging engine encountered an exception while starting. Exception:  
com.ibm.ws.sib.msgstore.PersistenceException:

**CWSIS1501E:** The data source has produced an unexpected exception:  
com.ibm.db2.jcc.b.SqlException: DB2 SQL error: SQLCODE: -552, SQLSTATE:  
42502, SQLERRMC: WPSADMIN;CREATE SCHEMA...

CWSID0035E: Messaging engine XYZ cannot be started; detected error reported during  
com.ibm.ws.sib.msgstore.impl.MessageStoreImpl start()

**Cause:** The messaging engine will not start because it is being asked to create its schema and tables in the database and is being refused due to a permission violation by the database.

**Solution:** Grant the needed permissions to the user ID specified in the data source. The database user ID that the messaging engine uses must have sufficient privilege to enable the messaging engine to create and access the data store tables. If you want the messaging engine to create the data store tables automatically, the messaging engine user ID requires additional privileges.

[http://www.ibm.com/support/knowledgecenter/SSAW57\\_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/rjm0650\\_.html](http://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/rjm0650_.html)

# Messaging Engine Startup Issues:

## CWSIS0002E followed by filestore full messages:

**CWSIS0002E:** The messaging engine encountered an exception while starting.

Exception: com.ibm.ws.sib.msgstore.PersistenceException: Unexpected exception caught starting persistent message store.

**CWSOM1042E:** ObjectStore=AbstractObjectStore(/<PATH to file store Log>/<some id>)/<some id>(ObjectStore) was asked to allocate space for ManagedObject=ManagedObject(null/null)/Constructed/<some id>(ManagedObject) when it was full.

**CWSIS1574E:** The file stores permanent store file is full.

CWSOM0003I: ObjectManager using logFile /<PATH to file store Log>/Log has shut down.

CWSID0035E: Messaging engine <ME NAME> cannot be started;

**Cause:** CWSOM1042E message indicates a space constraint for the file store has prevented the messaging engine from starting.

CWSIS1002E error with a nested error message of CWSM1042E and range of CWSIS1573E - CWSIS1575E also exists.

**CWSIS1573E:** The file store's log file is full.

**CWSIS1574E:** The file store's permanent store file is full.

**CWSIS1575E:** The file store's temporary store file is full.

**Solution:** Clean up any excess messages in queues or if necessary increase the size of the file store's log file or permanent store file or temporary store file. The CWSIS157\*E message indicates which file you need to update.

# Messaging Engine Startup Issues:

## ME startup fails with ORA-28000 error:

CWSIS0002E: The messaging engine encountered an exception while starting. Exception:  
com.ibm.ws.sib.msgstore.PersistenceException:CWSIS1501E: The data source has produced an unexpected exception:

java.sql.SQLException: **ORA-28000: the account is locked**

DSRA0010E: SQL State = 99999, Error Code = 28,000

**Cause:** The Oracle® account was locked

**Solution:** Unlock the account.

To unlock the account, run the following SQL query in Oracle using admin user.

```
ALTER USER <username> ACCOUNT UNLOCK GRANT CONNECT,RESOURCE to <username>
```

# Messaging Engine Startup Issues:

## **ORA-01031:insufficient privileges:**

CWSIS0002E: The messaging engine encountered an exception while starting.  
Exception: com.ibm.ws.sib.msgstore.PersistenceException: CWSIS1501E: The data source has produced an unexpected exception: java.sql.SQLException: **ORA-01031: insufficient privileges**

**Cause:** The user running the messaging engine does not have all the required privileges to access the datastore.

**Solution:** When using Oracle for the messaging engine datastore, please make sure that either the SchemaName and the userid specified in the Authentication Alias of the Data Source used by the messaging engine are same or if they are different the userid must have all the required privileges.

# Messaging Engine Startup Issues:

## CWSIS1501E, SQL30082N security issues:

CWSIS1501E, SQL30082N with database security error codes 08001 and -30,082.

CWSIS0002E: The messaging engine encountered an exception while starting. Exception:  
com.ibm.ws.sib.msgstore.PersistenceException: **CWSIS1501E**: The data source has produced an unexpected exception: java.sql.SQLException: **SQL30082N** Attempt to establish connection failed with security reason "24" ("**USERNAME AND/OR PASSWORD INVALID**"). SQLSTATE=08001  
DSRA0010E: SQL State = 08001, Error Code = -30,082

**Cause:** UserID or password is incorrect.

**Solution:** Verify the component managed authentication alias. The authentication alias must contain a valid user name and password combination that has appropriate access privileges for the database. This alias can be specified for a data source definition, or for a messaging engine definition.

# Messaging Engine Startup Issues:

## **CWSOM1017E: Unable to access log file:**

CWSIS0002E: The messaging engine encountered an exception while starting. Exception: com.ibm.ws.sib.msgstore.PersistenceException: Unexpected exception caught starting persistent message store:

**CWSOM1017E:**ObjectManagerState=ObjectManagerState( < Log file path> )/Stopped ManagedObject(0/0)/Ready/<id>(ObjectManagerState) caught exception=<Log file location> (Exception) trying to locate or create log file name={2}(String).

CWSID0035E: Messaging engine <ME\_NAME> cannot be started;

**Cause:** Messaging Engine is not able to access file system locations used by file store.

### **Solution:**

- 1) Incorrect path. File store locations provided do not exist.
- 2) User account is not authorized to read or write to the file system.
- 3) Some times NFS™ version 3 file system may create this type of problem because of its locking issues. NFS3 file system has some locking issues and we suggest to use NFS4 if you see any locking related issues.



# Messaging Engine Startup Issues:

## **CWSIS1535E: ME UUID does not match that found in the data store:**

In SystemOut.log:

CWSIS1538I: The messaging engine, ME\_UUID=ME2\_UUID, INC\_UUID=INC1, is attempting to obtain an exclusive lock on the data store.

CWSIS1545I: A single previous owner was found in the messaging engine's data store,  
ME\_UUID=ME1\_UUID, INC\_UUID=INC1

**CWSIS1535E: The messaging engine's unique id does not match that found in the data store.  
ME\_UUID=ME2\_UUID,ME\_UUID(DB)=ME1\_UUID**

**Cause:** The most likely cause of this is that someone deleted and recreated the ME without also deleting and recreating its message store. Another possible cause is that the ME was trying to get the lock on the data store owned by a different ME.

**Solution:** Ensure that the data source the ME is using references the correct database instance and that schemas being used by each messaging engine *are unique*. In WebSphere Application Server V6, V7, and V8 make sure that you have dropped data store tables after deleting a messaging engine.

# Messaging Engine Startup Issues:

## **CWSIS1519E - failed to obtain lock on cluster failover environment:**

CWSIS1538I: The messaging engine is attempting to obtain an exclusive lock on the data store.

CWSIS1519E: Messaging engine MyCluster.000-MyBus cannot obtain the lock on its data store, which ensures it has exclusive access to the data.

**Cause:** In a clustered environment when an ME loses its connection to its DB the HA manager will shutdown the ME on this JVM and move the ME to the other cluster member. When this happens the old lock on the database SIBOWNER table still exists. So the new ME instance is not able to get a fresh lock on the data store tables and ME unable to start on the other cluster instance.

**Solution:** Reduce the TCP KeepAlive timeout value at on the database server to ensure cleanup of the orphaned lock from the previous ME instance. This entire scenario is discussed in great detail in a previous WSTE, here:

[WebSphere Application Server - Service Integration Bus Messaging Engine Data Store Connectivity Problems and Solutions](#)

# Messaging Engine Startup Issues:

## ME Starts on 2 Different Servers in a Clustered Environment:

This problem is often discovered by the appearance of the following exception...

CWSIS1546I: The messaging engine, ME\_UUID=[329E8A0621345D24F](#), INC\_UUID=[47dd5495641dc6f03b](#), has lost an existing lock or failed to gain an initial lock on the data store.

Server 1 SystemOut.Log shows...

CWSIS1546I: The messaging engine, ME\_UUID=[329E8A0621345D24F](#), INC\_UUID=[47dd5495641dc6f03b](#), has lost an existing lock or failed to gain an initial lock on the data store.

Server 2 SystemOut.Log shows...

CWSIS1538I: The messaging engine, ME\_UUID=[329E8A0621345D24F](#), INC\_UUID=[512023701d9cefde](#), is attempting to obtain an exclusive lock on the data store.

In this case the issue is with HA Manager. Check whether HA manager is not enabled (by default it should be enabled). The HA Manager sorts out which ME to start on which server. If HA manager is enabled then check any communication issues between these 2 servers. This problem is not caused by SIB code.

# Messaging Engine Startup Issues:

## Slow Messaging Engine Startups:

If you are using persistent messages and you have a large number of these messages on destination (a queue or topic) and the Messaging engine is restarted for any reason the Messaging Engine may take a long time to startup. The reason for this is that during Messaging Engine restart each message must be read from the message store and recreated on the SIB destination (a process called Reconstitution).

In WAS 8.0.0.8 and V8.5.5.2 we have added a new custom property called '**printReconstatsEnabled**' so that an administrator can view a quick snapshot on the progress of messaging engine reconstitution in the SystemOut.log file. A message is logged in the SystemOut.log for every 90 seconds about the progress of the messages reconstitution. This feature is enabled by default.

If printReconstatsEnabled is set to false, then messaging engine startup information messages are not added to the SystemOut log.

**CWSIP0921I: Loaded 6 destinations, loading: Q1-Remote queue Point: 1, metadata loaded: 30,000/167,431**

**CWSIP0921I: Loaded 6 destinations, loading: Q1-Remote queue Point: 1, metadata loaded: 100,000/167,431**

PI05801: SIBUS TO OUTPUT STATUS MESSAGES DURING RECONSTITUTION PHASE OF ME STARTUP

<http://www.ibm.com/support/docview.wss?uid=swg1PI05801>

# Messaging Engine Startup Issues:

## Slow Messaging Engine Startups (continued):

If you are using WAS V8.5 you can greatly speed up message reconstitution using a new custom property named:

### **sib.processor.maxReconstituteThreadPoolSize**

This property can only be used if your machine is a multicore (multiple CPU) system and your database supports parallel reads by multiple threads (most do). Set this value equal to the number of CPUs (or CPU cores) on your system for parallel message reconstitution. This will significantly shorten Messaging Engine startup time.

# Messaging Engine Startup Issues:

## CWSIP0056E – Datastore corrupted:

**CWSIP0056E:** The link with name `com.ibm.ws.sib.processor.impl.DestinationManager` and UUID `<UUID>` cannot be recovered from data store.

`com.ibm.websphere.sib.exception.SIResourceException:`

`com.ibm.websphere.sib.exception.SIErrorException:`

`java.io.StreamCorruptedException: invalid stream header at  
com.ibm.ws.sib.processor.impl.BaseDestinationHandler.reco  
nstitute(BaseDestinationHandler.java)`

This exception often indicates a corrupt data store.

**Cause:** It is usually impossible to pinpoint the exact cause of data corruption unless it is something that can be easily recreated.

**Solution:** In most cases the only solution is to drop the database tables and recreate them.

# Application Connection Issues:

Sometimes even though an ME is up and running an application may be unable to connect to it for a variety of reasons. In most of the cases we see a CWSIT0006E exception which indicates the application is unable to connect to any of the specified bootstrap servers, preventing the client application from connecting to bus. This situation might be due to configuration problems, network problems, runtime issues or none of the specified bootstrap servers are currently available.

1. Runtime issues
2. Configuration issues
3. Authentication issues
4. Service unavailable exceptions.

# Application Connection Issues:

## **CWSIT0006E and CWSIA0241E:**

Application throws JMS exceptions CWSIT0006E and CWSIA0241E when trying to connect to a Service Integration Bus.

javax.jms.JMSEException: **CWSIA0241E**: An exception was received during the call to the method JmsManagedConnectionFactoryImpl.createConnection:com.ibm.websphere.sib.exception.SIResourceException: **CWSIT0006E**: It is not possible to contact a messaging engine in bus <bus\_name> at com.ibm.ws.sib.api.jms.impl.JmsManagedConnectionFactoryImpl.createConnection(JmsManagedConnectionFactoryImpl.java:225)

**Cause** : This can be caused due to network issues, stopped Messaging Engine, a wrong or misspelled bus name (case-sensitive) in the connection factory settings or, the Work Load Manager is not able to find the running messaging engine within its linger time.

**Solution**: The error messages CWSIA0241E and CWSIT0006E indicate that when the connection was attempted, there were no running Messaging Engines available on the specified bus. To see why the Messaging Engines were not available, check the SystemOut.log for applicable messages. Examine the SystemOut.log file starting at the time period when the application experienced the problem and search backward.



# Application Connection Issues:

## **CWSIT0006E and CWSIA0241E (continued):**

1) Along with these errors if you see any TCP errors or any networking related exceptions fix these problems. Check the network connection between the application and bootstrap server.

2) Check for the CWSID0016I which indicates status of the Messaging Engine activity. You may see messages like this:

```
SibMessage I [:] CWSID0016I: Messaging engine [mename]is in state Stopping.  
SibMessage I [:] CWSID0016I: Messaging engine [mename]is in state Stopped.
```

If ME is stopped then start the Messaging Engine.

3) With the above exceptions some times application receives a “ServiceUnavailableException” the most common reason for this is wrong providerEndpoints in the connection factory settings.

If you are using non-default bootstrap server settings then you need to configure ProviderEndpoints in the connection factory settings.



# Application Connection Issues:

## **CWSIT0006E and CWSIA0241E (continued):**

A bootstrap server listens on an endpoint that is defined by the combination of a comma-separated list of endpoint triplets.

Default is: **localhost:7276:BootstrapBasicMessaging**

- The host name of the host on which the bootstrap server is running
- A specific port that is either SIB\_ENDPOINT\_ADDRESS or, if security is enabled, SIB\_ENDPOINT\_SECURE\_ADDRESS
- A bootstrap transport chain

You can check this in the admin console or in resources.xml file. For Example:

```
bootstrapHost1:7278:BootstrapBasicMessaging,bootstrapHost2:7289:BootstrapSecureMessaging
```

After you modify this all application and messaging servers need to be restarted. Note, any modifications to the provider endpoint values should ALWAYS be done within the WAS Admin Console. Do NOT manually edit any XML files unless specifically directed to do that by IBM.

# Application Connection Issues:

## CWSIA0241E, CWSIT0019E and CWSIT0103E:

CWSIA0241E, CWSIT0019E and CWSIT0103E errors are returned to client . These errors are seen in the SystemOut.log even though Messaging Engine is running in one of the messaging cluster servers.

Javax.jms.JMSEException: **CWSIA0241E: An exception was received during the call to the method JmsManagedConnectionFactoryImpl.createConnection:**

com.ibm.websphere.sib.exception.SIResourceException: CWSIT0019E: No suitable messaging engine is available on bus <BUS\_NAME> that matched the specified connection properties {multicastInterface=none, connectionProximity=Bus, targetSignificance=Preferred, subscriptionProtocol=Unicast, targetType=BusMember, busName=<BUS\_NAME>}. Reason for failure: **CWSIT0103E: No messaging engine was found that matched the following parameters:** bus=<BUS\_NAME>, targetGroup=null,targetType=BusMember, targetSignificance=Preferred, transportChain=InboundSecureMessaging, proximity=Bus.

**Cause:** In a clustered environment it is the responsibility of the WLM to discover the running ME which matches the client request. If WLM is unable to find the running ME within the **sib.trm.linger** time of 3 seconds it throws the 'no ME found' exception back to client.

**Solution:** Increase the **sib.trm.linger** delay which increases the ME lookup time for WLM in clustering environment. This property can be specified in the sib.properties file.

# Application Connection Issues:

## Unresponsive Server / CWSIJ0042E:

You might see a CWSIJ0042E message in the log.

A connection from host <host\_name> has been terminated after it became unresponsive after a period of 7 seconds.

If a heartbeat fails (default wait is 7 seconds), the connection is terminated.

Check the server logs for hangs (**WSVR0605W**) or CPU Starvation detected (**HMGR0152W**) messages.

**Solution:** Increase the SIB heartbeat timeout by setting the following in the sib.properties file:

```
com.ibm.ws.sib.jfapchannel.HEARTBEAT_TIMEOUT=30
```

This increase the timeout to 30 seconds. The default is 7 seconds. This is often useful on very busy systems.

## CWSIV0775W during MDB initialization:

You might see a CWSIV0775W message in the log: Could not start messaging endpoint

This could be because application is initialized before SIBus completes the initialization.



# Application Connection Issues:

## CWSIV0775W, CWSIT0088E and CWSIT0103E:

WLM is unable to find the running ME, even though the requested ME is up and running

**CWSIV0775W:** The creation of a connection for destination <destination name> on bus <Bus\_name> for endpoint activation [com.ibm.ws.sib.ra.inbound.impl.SibRaStaticDestinationEndpointActivation @1780178 <active=true> <connections={}> <messageEndpointFactory=com.ibm.ejs.container.MessageEndpointFactoryImpl @aaf7b289> <endpointConfiguration=[ <JmsJcaActivationSpecImpl.this=[.....] failed with exception com.ibm.websphere.sib.exception.SIResourceException:**CWSIT0088E: There are currently no messaging engines in bus YOUR\_BUS running.** Additional failure information: **CWSIT0103E: No messaging engine was found that matched the following parameters:** bus=<BUS\_name>, targetGroup=null, targetType=BusMember, targetSignificance=Preferred, transportChain=InboundBasicMessaging, proximity=Bus..

...  
DiscoveryRcv W **DCSV1115W:** DCS Stack <server\_name>: Member <server\_name> **connection was closed. Member will be removed from view.** DCS connection status is Discovery|Ptp, receiver closed.

**Cause:** In clustering environment ME is running on a server that is not part of the local core group, and there is no bridge between the local core group and the one where the desired ME is running, then the WLM will not be able to find the ME, even though it is running. In other words, the servers are not communicating with each other properly and as a result one server cannot see the ME on another server.

**Solution:** Add all servers to the same core group, or configure a bridge between the core groups if the servers need to be members of a different core group. Also check for any networking issues between servers or nodes.

# Message Store Problems

Message store problems can be broken down into 3 categories:

1. Data store problems
2. File store problems
3. Configuration problems

Data store problems are problems involving the interaction between a messaging engine and its database tables.

File store problems are problems involving the interaction between a messaging engine and its file store files.

Configuration problems are problems that usually involve an incorrect use of a message store by more than one messaging engine, incorrect data source or schema configuration or are the result of manually editing data inside the message store.



# Message Store Problems

## Data store problems:

1) Most data store problems become apparent during messaging engine start up. The most common message indicating a problem is a *repeating* CWSIS1593I informational message. For example:

**CWSIS1593I:** The messaging engine, ME\_UUID=307BE55B4A98BFE1, INC\_UUID=3AA13AA12707C798, **has failed to gain an initial lock on the data store.**

This usually happens after a server has been killed by a user. The CWSIS1593I message means that the data store tables the messaging engine needs to connect to during startup have an old lock on them. This old lock prevents the messaging engine from getting a new lock on the tables. A previous WebSphere Technical Exchange presentation provides all of the details about how this situation arises. You can find that presentation here:

Service Integration Bus Messaging Engine Data Store Connectivity Problems and Solutions:

<http://www.ibm.com/support/docview.wss?uid=swg27020333>



# Message Store Problems

**Solution:** The solution to this problem is to set the TCP KeepAlive value *on the database server* to a value less than 15 minutes. We usually recommend 3 to 5 minutes. It is NOT necessary to set this on the WAS server. Details can be found in the other presentation.

Also, if you are running WAS version 8.5 or higher this problem can usually be avoided entirely by enabling the “Restrict long running locks” feature in the message store configuration panel. Instructions for enabling this feature can be found [here](#).



# Message Store Problems

2) The connection from the messaging engine to the database ends or terminates unexpectedly. You will likely see a **J2CA0056I** message in the SystemOut.log file. This will typically appear with a **java.net.SocketException: Connection reset error**.

When the connection between a messaging and its data store ends unexpectedly the messaging engine will usually failover to another server in the messaging cluster and try to get a new lock on it's data store. That will result in the situation just described in problem number 1 in the previous 2 slides.

**Solution:** Again, in these situations the TCP KeepAlive parameter is the best solution unless you are running WAS 8.5 or higher, in which case the “Restrict long running locks” feature is the best solution.



# Message Store Problems

## File store problems:

1) By default SIB uses persistent messages, which means that each message in the JMS system is stored in the permanentStore file store file. By default this file is set to a maximum size of 500 MB but this may not be large enough in some environments. If this file fills up JMS messaging will stop (likely causing application problems) and the following message will appear in the SystemOut.log file:

**CWSIS1574E:** The file store's permanent store file is full.

**Solution:** The obvious solution is to increase the size of this file in the message store configuration to accommodate your messaging load. *Note that this exception often indicates that there are a large number of messages on a JMS queue. If messages are stacking up on a queue that indicates a problem with a message consumer application.*

# Message Store Problems

2) A very similar exception can occur for the Log file store file:

**CWSIS1573E:** The file store's log file is full.

This second exception for the Log file suggests that you may be using very long running transactions in your applications, such as transactions that include thousands of messages in a single transaction, or you may be using very large JMS messages.

**Solution:** Try reducing the size of your transactions or the size of your messages if possible. You can increase the size of the Log file but *long running transactions should still be avoided whenever possible.*

# Message Store Problems

3) The messaging engine fails to start and you see the following exception in the SystemOut.log file:

**CWSIS1579E:** The file stores log file is locked by another process.

**Cause:** There can be several causes for this. On a Windows system a virus scanner or other third party application may have the file locked. On a Unix®/Linux® system the more likely cause is a known problem with file locking in NFS version 3.

**Solution:** On a Windows system you should try using Process Explorer from Microsoft to determine which process has a lock on the Log file. If you are running Unix/Linux and you are using NFS version 3 you should upgrade to NFS version 4 because it uses a much better lease based file locking mechanism that avoids this problem.

Your OS administrator may be able to manually unlock the file. In a worst case scenario you can rename all 3 of the file store files and restart the messaging cluster. This will create a new set of files that should be successfully locked by the messaging engine.



# Message Store Problems

4) Many messages have been removed from a JMS destination but the file store permanentStore file does not shrink in size.

**Solution:** This is not actually a problem. The file size will shrink only after every 5,000 file store operations (transactions). The file does not resize after every message for 2 reasons:

1. It is assumed that if you reached a certain size once you are likely to do it again, so the file is not resized.
2. There is a great deal of overhead required to resize the file after each and every message. The memory and CPU resources required for this would be better utilized for other purposes.



# Message Store Problems

## Configuration problems:

1) **Not Authorized** exception (java.sql.SQLException) returned from the database during messaging engine start up.

**Solution:** Make sure the user ID and password specified in the Authentication Alias of your Data source is correct.

2) After dropping the SIB tables new tables are not automatically created during messaging engine start up.

**Solution:** Enable the “Create tables” option in the Message store configuration panel in the WAS Admin Console. *Some products disable this feature during installation.* The default setting is Enabled.

# Message Store Problems

3) During messaging engine startup the following message appears in the SystemOut.log and the startup fails:

**CWSIS1535E:** The messaging engine's unique id does not match that found in the data store. ME\_UUID=E52AB8AEE2F-05510, ME\_UUID(DB)=A15FEDA-F0185937C

NOTE: The UUID values above will be different from yours but the format will be the same.

**Cause:** This problem occurs if you delete and recreate a messaging engine without also deleting and recreating its associated message store. This can also happen if 2 messaging engines are configured to share the same message store (data store or file store). This is not allowed!

**Solution:** If you are using a data store DROP all of the SIB tables owned by the messaging engine and restart the messaging engine. If you are using a file store change the names of the 3 file store files and restart the messaging engine. The filenames are “Log”, “temporaryStore”, and “permanentStore”. Also make sure each messaging engine has it's own, unique message store.

# Message Store Problems

4) Another common problem is the attempt to use an incorrect database driver. This is often the case when Oracle is being used and results in a DSRA7042W warning message. For example:

**DSRA7042W:** Oracle does not support the use of version 10 of their JDBC™ driver with the version of the Java runtime environment that is used by the application server.

This means that you are using an incorrect JDBC driver.

**Solution:** You should use the 11g driver *even if you are running a version of Oracle that is earlier than version 11g*. Change the driver in the WAS Admin Console and restart the messaging cluster for the change to take effect.



# Message Store Problems

4) This message is sometimes seen as a popup message in the WAS Admin Console when a Messaging Engine attempts to start, but there is no corresponding CWSI type exception message in the SystemOut.log file:

The messaging engine **YOUR\_ME\_NAME** cannot be started as there is no runtime initialized for it yet, retry the operation once it has initialized. For the runtime to successfully initialize the hosting server must be started, have its 'SIB service' already enabled, and dynamic configuration reload enabled. If this is a newly configured messaging engine and it is the first messaging engine to be hosted on this server, then it is most likely the 'SIB service' was not previously enabled and thus the server will need to be restarted. The messaging engine runtime might not be initializing because of an error while trying to start, examine the SystemOut.log of the hosting server to check for error messages indicating the problem.

# Message Store Problems

There are 4 known causes for this problem, although in almost every case the cause is a problem with the message store:

## Causes:

- 1) Servers in messaging cluster are not running.
- 2) SIB Service is not enabled
- 3) Message store problems
- 4) Incorrect core group policy. See this link:

<http://www.ibm.com/support/docview.wss?uid=swg21591208>

# Message Store Problems

## Avoiding Problems:

### Data Stores:

- Use the correct JDBC driver
- Ensure your Data source has the correct user ID and password specified in the Authentication Alias
- *Make sure each messaging engine has a unique schema name in the Message store configuration panel. Messaging engines cannot share a schema name!*
- Avoid running tooling, such as runstats or similar tools, against the SIB tables. This *will* cause problems.

### File Stores:

- Make sure each messaging engine has it's own, unique set of file store files. This means using a unique file store directory path for each messaging engine.
- *Put your file store on a shared file system* so that no matter what node/server your messaging engine is running on it can connect to the file store.
- Avoid backing up file store files while the messaging engine is actively using the files. This will create a file locking problem.

# Message Flow Problems

Message flow problems are usually characterized by unexpected message behavior on a JMS queue. We will look at the 4 most common situations:

1. Messages are not being consumed from a destination
2. Messages are consumed very slowly from a destination
3. Messages are sent to the SYSTEM\_Exception destination
4. Messages stuck in a PENDING\_ACKNOWLEDGEMENT state

In the vast majority of cases these problems are driven by application issues.

# Message Flow Problems

## **Messages are not being consumed from a destination:**

This behavior is almost always driven by application behavior. In most cases the cause is a consumer Message Driven Bean that is either hung or is waiting for a long time on some external component it is connected to such as a database or a web service.

If messages are stuck on a queue please use the following Websphere Technical Exchange presentation for guidance on what to look for and how to move forward with the problem:

## **Service Integration Bus and Stuck Messages**



# Message Flow Problems

## **Messages are consumed very slowly from a destination:**

This behavior can occur when you have very large JMS messages or when the message consumer code (your application) takes a long time to process a message (even small messages). You can measure how long each message is taking to process by enabling the following trace on the server where the messaging engine is running:

```
*=info:SIBProcessor=all:SIBMessageTrace=all:SIBCommunications=all:SIBRa*=all:SIBJmsR  
a*=all: SIBJFapChannel=all:SIBMfp=all:EJBContainer=all
```

# Message Flow Problems

In the trace look for entries such as this:

```
[4/5/12 14:47:09:238 EDT] 00000fc MDBWrapper > onMessage Entry
... more lines ...
[4/5/12 14:48:13:518 EDT] 00000fc MDBWrapper < onMessage Exit
```

Notice that each entry is on the same thread (**00000fc**). The first line shows the **onMessage()** method being called. This is the entry point into the Message Driven Bean and this is where SIB passes a message to the bean. The second line shows the exit from **onMessage()**. *Now note the difference in the timestamps.* This shows that the message took about 1 minute and 4 seconds to process. This is very slow. If you see this behavior in a trace the next step is to gather Javacores from the server where the MDB runs and analyze the bean code. Details about exactly what to do and what to look for can be found here:

## Service Integration Bus and Stuck Messages

# Message Flow Problems

## Messages are sent to the **SYSTEM\_Exception** destination:

Messages are sent to the **SYSTEM\_Exception** destination when they cannot be processed or routed normally. Most of the time a message is in this queue because a Message Driven Bean is having a problem processing the message. By default SIB will deliver a message to an MDB a maximum of 5 times. If all 5 attempts to process the message fail the Messaging Engine will move the message to the Exception destination so that the next message on a queue can be processed.

When a message is sent to the Exception destination some information is added to the message header to help determine why the message could not be processed or routed. If you list the messages on the Exception destination and then click on one of the messages the header information for that message will be displayed. In the list of header fields check the **Exception destination reason** field to see why the message was sent to the Exception destination.



# Message Flow Problems

## Messages are sent to the `SYSTEM_Exception` destination:

The usual reason is that the maximum number of delivery attempts had been reached, but sometimes a more specific reason is given and this can often be very helpful in determining the cause of a problem.

SIB does not include an automatic mechanism for handling (moving or processing) these messages. You could write an application to move the messages to their original destination to try and process them again but IBM has an application that will do this and much more for you. We recommend using the Bus Destination Handler tool for this purpose. You can download the tool here:

## Bus Destination Handler

Note that this tool is provided **AS-IS** (no support) but it works very well.



# Message Flow Problems

## Messages stuck in a **PENDING\_ACKNOWLEDGEMENT** state:

Make sure you have the following APARs installed, *especially if you are running the IBM Connections product*:

**PM51310**: IN SERVICE INTEGRATION BUS, PUBLISH/SUBSCRIBE MESSAGES REMAIN IN "PENDING ACKNOWLEDGMENT MODE" STATE ON REMOTE PUBLICATION POINTS

**PM81521**: MESSAGES STUCK IN PENDING ACKNOWLEDGEMENT STATE IN WEBSHERE APPLICATION SERVER SERVICE INTEGRATIONS BUS DESTINATION

**PM56596**: MESSAGES BUILD UP ON PUBLICATION POINT AND DO NOT FLOW TO SUBSCRIBER MESSAGING ENGINE

**PM62615**: MESSAGES BUILD UP ON WEBSHERE APPLICATION SERVER V7.0 SERVICE INTEGRATION BUS TOPICSPACE

**PM71430**: PUBSUB MESSAGES STUCK ON WEBSHERE APPLICATION SERVER SERVICE INTEGRATION BUS PUBLICATION POINT AFTER DELIVER TO



# Message Producer/Consumer Problems

Problems directly involving message producers or consumers almost always fall into one of the following categories:

1. Producer Transaction Size
2. Consumer Scaling
3. Consumer Application Performance

Of these 3 categories message consumer performance is, by far, the most common problem. However, producer transaction size can also introduce performance issues.

# Message Producer/Consumer Problems

## Producer Transaction Size:

If your producer application sends very large numbers of messages within a single transaction it will flood the JMS system with data. This can cause OutOfMemory conditions. Also, none of those messages will be available for your consumer application until the commit() call is made to close the transaction. This means your consumer application may have to wait for long periods of time before it can do any work, which is inefficient.

Long running transaction from Producer applications can also cause the Log file full exceptions discussed earlier.

**Solution:** Avoid long running transactions. Keep transactions small and manageable. This will speed up message processing and it will make recovery from any problems much faster. It will also conserve memory on the heap.

# Message Producer/Consumer Problems

## Consumer Scaling:

One of the most common conditions seen in JMS is the accumulation of messages on a JMS queue. Many times this can be solved by scaling your consumer applications to increase message processing capacity. In general this can be done in 2 ways (if you are using Message Driven Beans for your consumers):

1. Increase the “**Maximum concurrent MDB invocations per endpoint**” value in the Activation Specification used by your MDB. The default value is 10 but we have seen it as high as 50.
2. Increase the number of servers in your application cluster. This can increase the number of MDBs running against the queue.

More information about scaling for efficiency can be found here:

[Ask the Experts Replay: Service Integration Bus Scalability Best Practices](#)



# Message Producer/Consumer Problems

## Consumer Application Performance:

Consumer application code or performance problems are extremely common and in most cases are the first thing that should be examined. Symptoms of performance problems include messages stuck on a queue in a Locked state or messages that are processed very slowly.

**Solution:** There are many possible causes for this behavior but the most important action that can be taken is to gather a set of javacores from the server where the consumer application runs. The javacores will reveal exactly where any delays are in the application code. Details on what to look for and how to interpret a javacore can be found here:

## Service Integration Bus and Stuck Messages



# Summary

In this presentation we have covered the following problems and their solutions in details:

- Messaging Engine start up issues
- Application connection issues
- Message store problems
- Message flow problems / stuck messages/message pileup issues
- Most common producer / consumer issues

# Connect with us!

## 1. Get notified on upcoming webcasts

Send an e-mail to [wsehelp@us.ibm.com](mailto: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](mailto:wsehelp@us.ibm.com)





# Open Lines for Questions



# 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](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>