

## Chapter 12 Extension (Contents added to the existing section)

### Service Integration Bus and High Availability Manager

#### Configure Highly Available Messaging Engine with Workload Sharing – Contents added are highlighted in Green

As you saw in an earlier task, the default SIBus member configuration doesn't support workload sharing because only one messaging engine is active to handle the traffic for that member. For many applications, one message engine will be enough to handle the messaging load. One reason to use a single messaging engine is to more easily maintain message sequence. For scenarios requiring high-volume message processing, you can change the default configuration by adding additional messaging engines to the cluster to support workload sharing. Figure 12-9 shows this architecture.

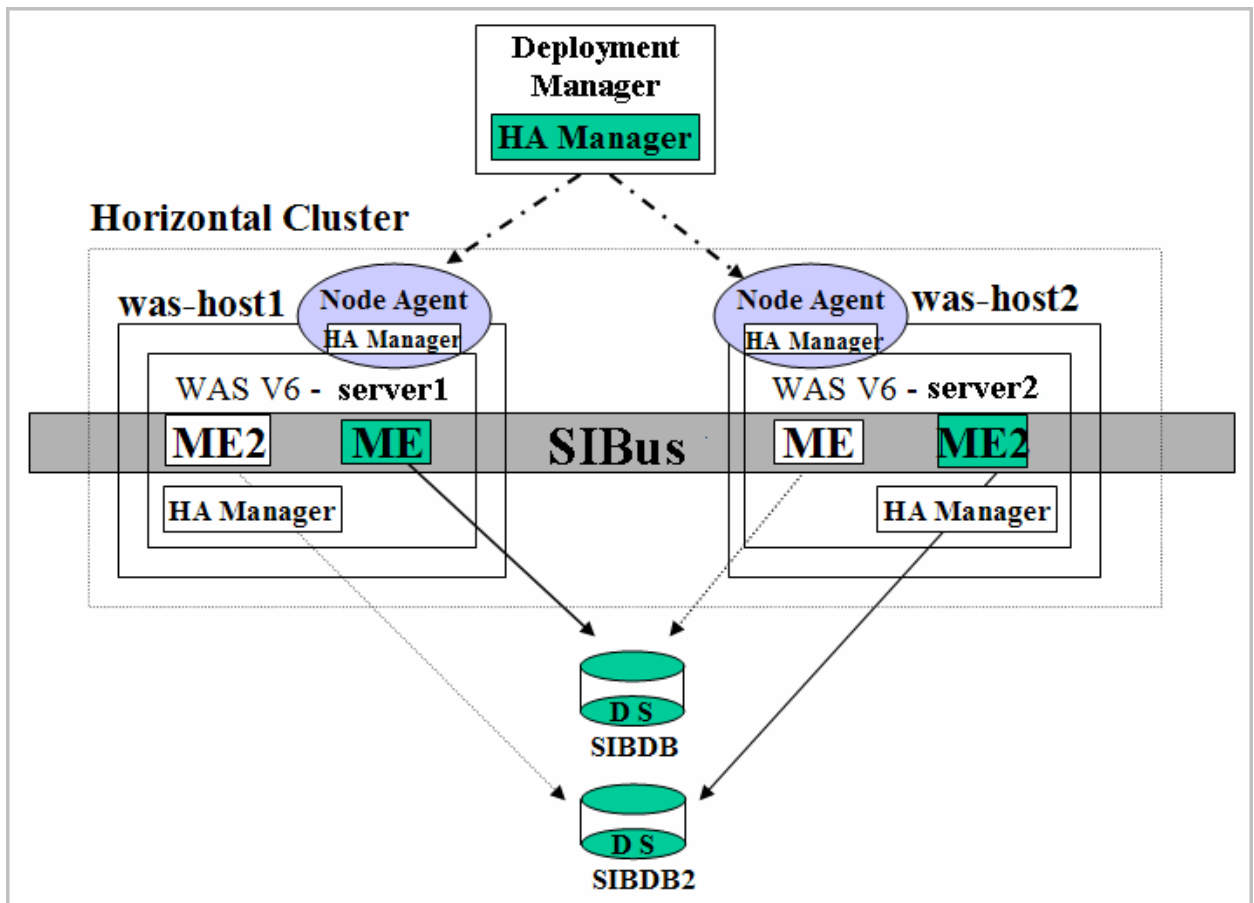


Figure 12-9: Highly available messaging engine with workload-sharing architecture

To set up this environment, complete the following steps to create a data store to persist the messages of a messaging engine on the second cluster member.

1. Create a second database to persist the messages. In the example, we create a second database (SIBDB2) on DB2.
2. Create the data source jdbc/sibus2, and verify the data-source database connection.
3. Navigate to **Service integration|Buses|yourbus|Bus members|yourmember**, and click the **Add messaging engine** button. Figure 12-10 shows this panel for the HorizontalCluster member.

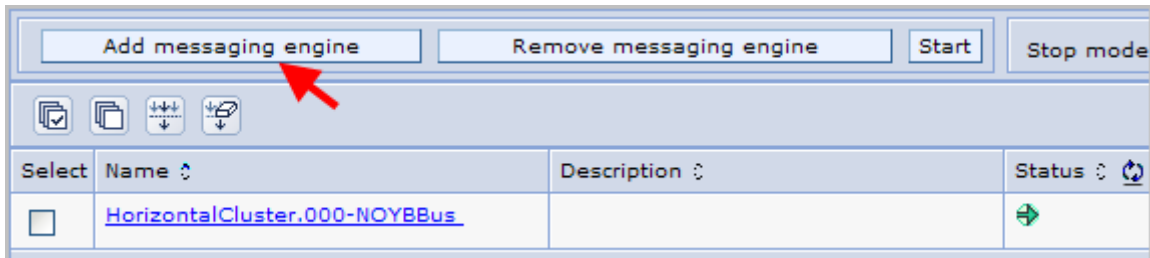


Figure 12-10: Adding a messaging engine

4. On the resulting panel (Figure 12-11), enter the name of the second data source and the J2C authentication alias you created for access to the SIBus database. Leave all other values at the defaults. Make sure the **Create tables** check box is selected. Click **OK**, and save the configuration.
5. Restart the cluster if it's practical to do so. Otherwise, start the new messaging engine on the cluster by navigating to **Service integration|Buses|yourbus|Messaging engines**, selecting the new message engine's check box, and clicking **Start**.
6. Use the instructions you followed earlier to verify the configuration and failover.

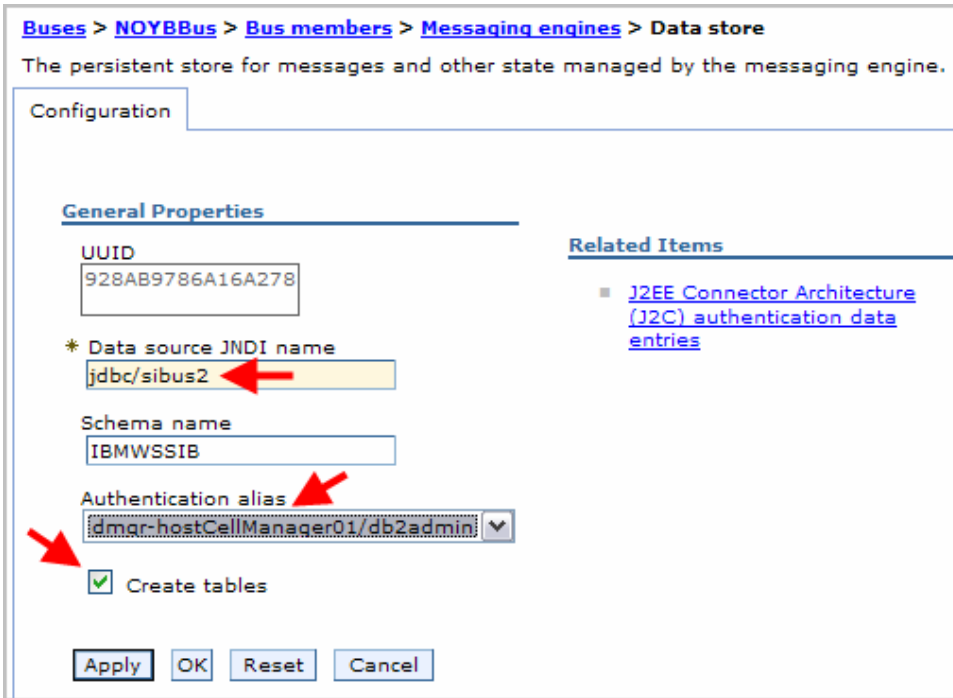


Figure 12-11: Creating the second data store

As shown in the screenshot below the second messaging engine will be created using the naming convention of **ClusterName.XXX-BusName** (HorizontalCluster.001-NOYBBus in our sample configuration). Where XXX represents the numeric value in the increasing order (000, 001 etc.) representing the number of the messaging engine in the cluster. You will be using the names of these messaging engines while configuring the match criteria for the new high availability policies you will create later in this section. Navigate to **Service Integration|Buses|yourbus|Messaging engines** to look at the names of the messaging engines in your configuration.

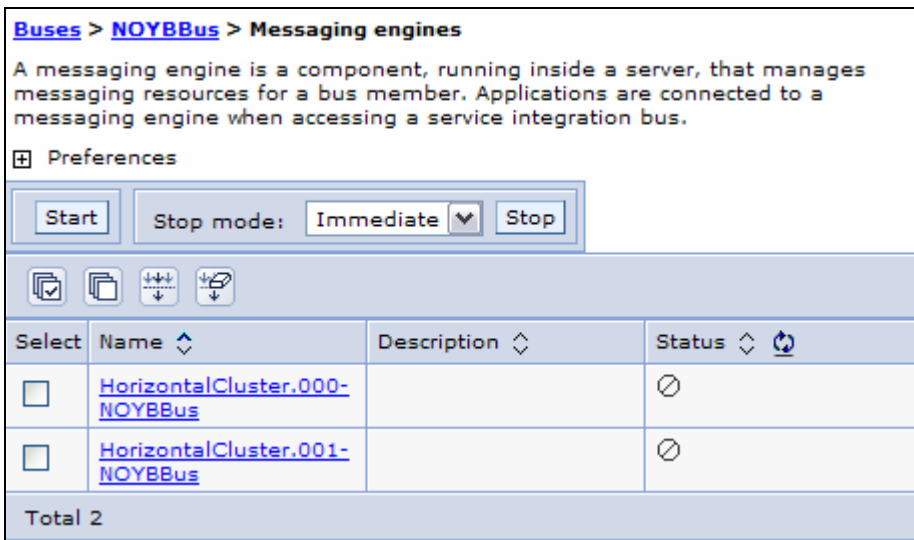


Figure 12-11a: Naming convention of messaging engines in a cluster environment

After adding and starting the additional messaging engine using the instructions above with the default configuration, you will find that both the active messaging engines (ME and ME2) are instantiated on the same server depending on which cluster member starts first (server1 or server2) as shown in the diagram below. You can verify this by looking at the log files (SystemOut.log for each cluster member) mentioned in the earlier section. Our intention is to distribute the active messaging engines (MEs) across both of the cluster members to attain a true HA/WLM environment for messaging engines. Even though the diagram shown below shows that both active MEs instantiated on server1, they might appear on server2, if server2 is the server that starts first.

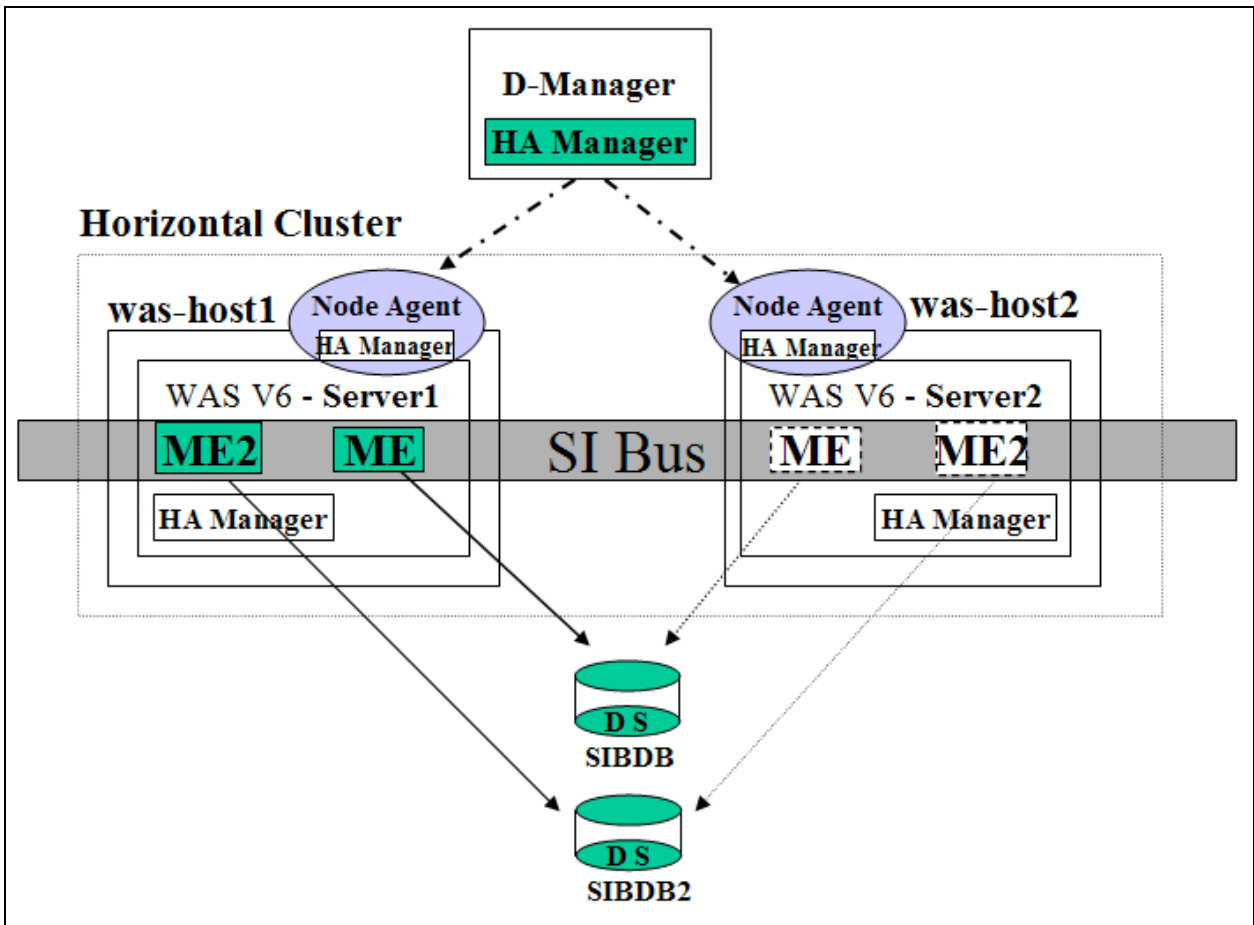


Figure 12-11b: Highly available messaging engine with the default configuration

In order to distribute the active messaging engines across the cluster members as described in Figure 12-9, you need to create two new high availability policies related to the messaging engines. Refer to the High Availability Manager section later in this chapter for more information about high availability policies. Perform the following tasks to create the new policies:

1. Create the first policy to start the first messaging engine (ME) always on the first cluster member.
  - a. Navigate to **Servers|Core groups|Core group settings|DefaultCoreGroup|Policies** and click **New** to create a new policy.

- b. Select **One of N Policy** and click on **Next**.
- c. Name the policy and click on **Apply**. We used **ME\_ON\_MEMBER\_1** as the name in our sample configuration. You will see the options under the **Additional properties** section will be enabled after clicking on **Apply**.
- It's usually a good idea to select the checkboxes for **Fail back** and **Preferred Servers only** as other servers are added to the core group there are likely to be servers where you don't want the MEs to run. Refer to the **High Availability Manager** section later in this chapter for more information about these options.

The screenshot shows a configuration window with two main sections: 'General Properties' and 'Additional Properties'. In the 'General Properties' section, the 'Name' field contains 'ME\_ON\_MEMBER\_1', the 'Policy type' is 'One of N policy', and the 'Description' is 'Policy to start ME on the first cluster member'. The 'Is alive timer' is set to '0 seconds'. There are three checkboxes: 'Quorum' (unchecked), 'Fail back' (checked), and 'Preferred servers only' (checked). At the bottom are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'. The 'Additional Properties' section on the right has three expandable links: 'Custom properties', 'Match criteria', and 'Preferred servers'.

Figure 12-11c: Creating a new policy

- d. Select the **Match criteria** link and click on **New** on the resulting screen.
- e. On the criterion configuration screen enter the **Name** as **type** and **value** as **WSAF\_SIB**. Click on **OK** and **save** the configuration.

Core groups > DefaultCoreGroup > Policies > ME\_ON\_MEMBER\_1 > Match criteria > New

Specify an arbitrary name and value pair of data, where the name is a property key and the value is a string value.

Configuration

**General Properties**

\* Name  
type

\* Value  
WSAF\_SIB

Description  
All the MEs in the Bus

Apply OK Reset Cancel

Figure 12-11d1: Specifying the first match criteria for the first policy

- f. Click on **New** again to create another match criterion under this policy.
- g. On the criterion configuration screen enter the **Name** as WSAF\_SIB\_MESSAGING\_ENGINE and **value** as name of the first messaging engine in your configuration. In our sample configuration it was HorizontalCluster.000-NOYBBus as shown in the screenshot below. Click on **OK** and **save** the configuration.

Core groups > DefaultCoreGroup > Policies > ME\_ON\_MEMBER\_1 > Match criteria > New

Specify an arbitrary name and value pair of data, where the name is a property key and the value is a string value.

Configuration

**General Properties**

\* Name  
WSAF\_SIB\_MESSAGING\_ENGINE

\* Value  
HorizontalCluster.000-NOYBBus

Description  
To start ME on server1

Apply OK Reset Cancel

Figure 12-11d2: Specifying the second match criteria for the first policy

After configuring the two match criterions, Match criterion page should look similar to the screenshot given below.

Core groups > DefaultCoreGroup > Policies > ME\_ON\_MEMBER\_1 > Match criteria

Define the match criteria for the policy. Match criteria consists of name and value pairs of data, where the name is a property key and the value is a string value.

Preferences

New Delete

Select	Name	Value	Description
<input type="checkbox"/>	type	WSAF_SIB	All the MEs in the Bus
<input type="checkbox"/>	WSAF_SIB_MESSAGING_ENGINE	HorizontalCluster.000-NOYBBus	To start ME on server1

Total 2

Figure 12-11: Two match criterions of the first policy

- h. Select the **Preferred Servers** link under the **Additional properties** section for this policy (ME\_ON\_MEMBER-1).
- i. Select the **cluster member one** (server1) and click on **Add >>** to move this cluster member as the **first** preferred server.
- j. Select the **cluster member two** (server2) and click on **Add >>** to move this cluster member as the **second** preferred server. This means this messaging

engine will be started on this server only if the first cluster member is not started or has failed.

k. Click on **OK** and **save** the configuration

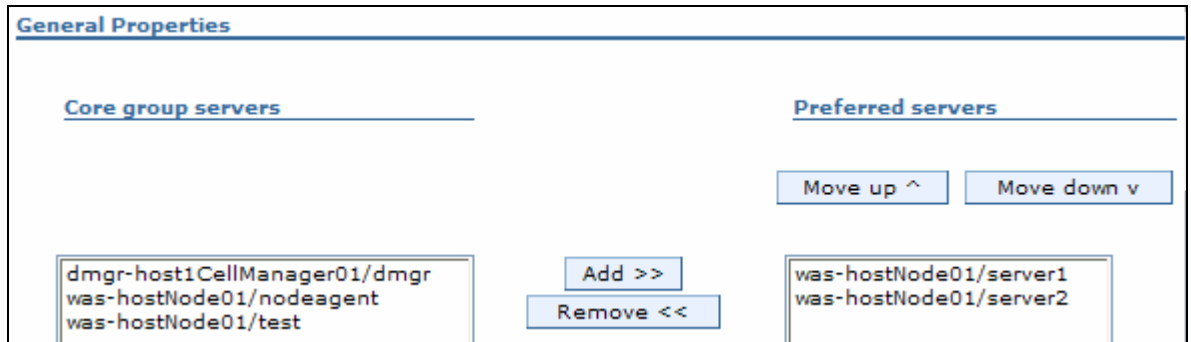


Figure 12-11e: Specifying the preferred servers and their order for the first policy

2. Create the second policy to start the second messaging engine (ME2) always on the second cluster member using the same instructions given above with the following variations:

- a. Provide a different policy name. We used ME\_ON\_MEMBER\_2 as a name for the second policy.
- b. For the second **Match criteria** enter the **Name** as WSAF\_SIB\_MESSAGING\_ENGINE and **value** as the **second** messaging engine in your configuration. In our sample configuration second messaging engine was HorizontalCluster.001-NOYBBus as shown in the screenshot below.
- c. For the **Preferred Servers**, **first** select the second cluster member (server2) and then the second cluster member (server1) in the order specified as shown in the screenshot below.

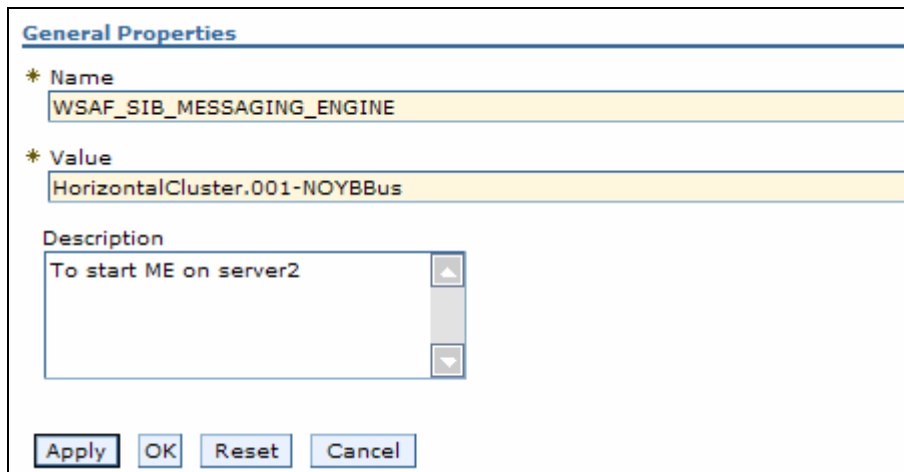


Figure 12-11f: Specifying the second match criteria for the second policy

c. For the **Preferred Servers**, **first** select the second cluster member (server2) and then the second cluster member (server1) in the order specified as shown in the screenshot below.



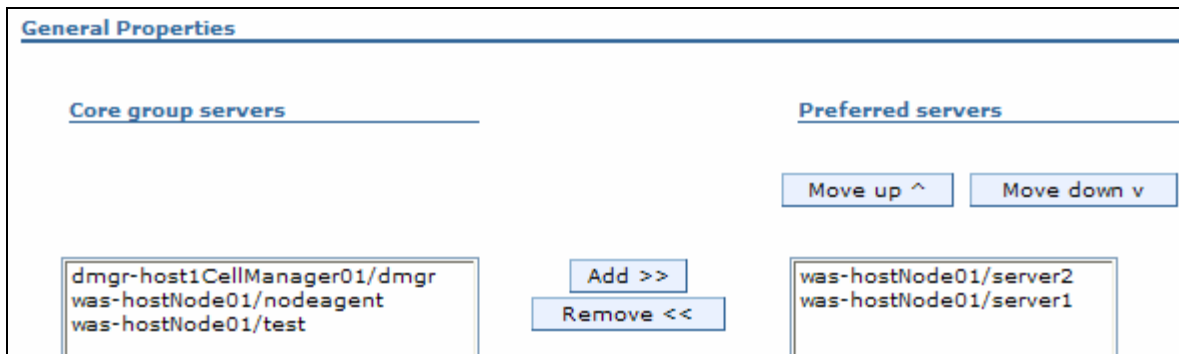


Figure 12-11g: Specifying the preferred servers and their order for the second policy

- d. Click on **OK** and **save** the configuration.
- 3. Restart the cluster and verify that the messaging engines (ME1 and ME2) will be started on both the cluster members (server1 and server2) as configured looking at the log files using the instructions provided earlier in this chapter.

-30-  
-30-