Deploy InfoSphere MDM Collaborative Edition onto a cluster, Part 1: Strategies for mixed clustered topologies on an application server

A step-by-step guide

Sachin Prasad (sachin.prasad@in.ibm.com)  
Solution Architect  
IBM

Abhijith Boompelly (abhijith.boompelly@in.ibm.com)  
Quality Engineer  
IBM

Srinath Shenoy (shenoysri@in.ibm.com)  
Engineering Manager  
IBM

This two-part tutorial shows how to set up a typical IBM® InfoSphere® Master Data Management (MDM) Collaborative Edition V11 clustered environment. This first part describes how to deploy InfoSphere MDM Collaborative Edition V11 onto a cluster. Follow along step-by-step with detailed examples of several clustering strategies. Also learn about the required configuration to enable the IBM HTTP Server for caching and load balancing the cluster nodes.

Introduction

This two-part tutorial is designed for developers, administrators, and transition engineers who want to set up a typical IBM InfoSphere Master Data Management Collaboration Edition V11 clustered environment. There are two ways to deploy InfoSphere MDM Collaborative Edition (MDM CE) onto a cluster:

- Deploying MDM CE on an application server and then converting it into a cluster.
- Deploying MDM CE directly on a computer cluster.

This first part of the tutorial covers the first mode of deployment whereby MDM CE is first installed and then manually clustered using mixed topology deployment. This article is based
on field experience gathered while installing and resolving clustering issues. It is intended as an additional reference to documentation on basic MDM CE installation in the Information Center (see Resources).

Part 2 explores deploying MDM CE directly on an existing cluster. It provides details on MDM CE fix pack upgrades over the cluster. It also briefly describes some advanced topics such as setting up dynamic clusters and using IBM Edge software as the front end of the cluster.

**In this tutorial**

The MDM CE V11 installer is based on IBM Installation Manager (IM) software. (IM replaces the InstallShield MultiPlatform (ISMP) based installer used in earlier versions.) IM provides an efficient mechanism for maintaining software packages with features that enable you to install, update, modify, and uninstall software. Get assistance for each version from the IBM IM Information Center.

This tutorial provides step-by-step instructions to use IM to install MDM CE V11 and to cluster the instance into a mixed topology. The tutorial also gives steps for setting up IBM HTTP Server (IHS) and Plug-in installation, which provide a front end for the cluster for high availability and load balancing. The tutorial ends with recommendations and considerations for other possibilities for setting up clusters.

The tutorial is divided into the following sections:

- Introduction to clustering with WebSphere® Application Server
- Overview of topology implemented in this article.
- Overview of MDM CE clustered environment deployment steps
- Installing stand alone MDM CE V11
- Preparing the cluster
- Configuring MDM CE on the cluster
- Installing IHS and Plug-in
- Configuring IHS for cluster
- Verifying the installation

**Prerequisites**

The following prerequisites are assumed:

- Knowledge of network deployment concepts of WebSphere Application Server
- A correctly configured instance of IBM IM
- A correctly configured instance of WebSphere Application Server Network Deployment
- A correctly configured instance of IBM DB2® for MDM CE

**Configuration used for the installation**

- Red Hat Enterprise Linux™ 6.2 (Santiago) servers
• IBM WebSphere Application Server Network Deployment V8.5.0.2
• IBM DB2 V10.1
• IBM Installation Manager V1.6
• InfoSphere MDM CE V11 install bundle
• Network File System (NFS) setup to share the MDM CE home directory

Clustering WebSphere Application Servers

To set up WebSphere clusters, you have to use WebSphere Application Server Network Deployment edition (WAS ND). The Base edition does not support clusters.

A cluster is a logical grouping of application servers that behave as a single server. The application servers that belong to a cluster are called its members. The application servers continue to run within the runtime environment of the node, but they operate jointly within the cluster as a single, virtual, distributed application server. The workload of the cluster is shared among the individual servers that make up the cluster. If one server experiences a fault and is disabled, the other servers can take over and redistribute the load. The overall system remains available. Clients accessing the system might never know that a server experienced any difficulties.

When deciding how to configure a WebSphere Application Server Network Deployment cluster, there are two strategies you can use: vertical clustering and horizontal clustering (or a combination of the two).

Vertical clustering

Deploying several application servers on a single physical computer is called vertical clustering because you stack the application servers onto a single physical computer vertically.

Use vertical clustering to take advantage of a machine with multiple CPUs and large amounts of memory to get maximum throughput. Sometimes the JVM itself can be the bottleneck and, because each application server is its own JVM, the simplest way to increase throughput is to have more JVMs running. And, because system faults are not always due to hardware problems, having more than one application server on each computer further increases the system’s fault tolerance (and hence availability) without additional investment in hardware. Vertical clustering is a simple and inexpensive way to achieve more performance from the system.

The frequently recommended method for scaling vertically is to have a single WebSphere node to host multiple servers. Though there can be more than one node in one computer, this is generally not recommended because each node runs its own node agent; having multiple node agents running is considered overhead. Figure 1 shows a simple vertical cluster.
Figure 1. Vertical cluster

Horizontal clustering

Horizontal clustering involves adding more computers to a cluster. You spread the cluster out broadly across several computers horizontally.

With horizontal clustering, multiple application servers are deployed on different physical computers, which provides an easy and convenient way of scaling beyond hardware limitations of a single computer. If a computer fails, the remaining computers are unaffected and the application can still be served by another cluster member. Horizontal clustering also increases the scalability of the system by distributing the processing across several CPUs. Typically, each physical computer comprises one node, so generally, you can say each computer is a cluster node. When people think of clustering, they generally think of horizontal clustering. Figure 2 shows a horizontal cluster.

Figure 2. Horizontal cluster

Mixed cluster

A mixed cluster topology employs a mix of vertical and horizontal clustering to meet performance requirements with the available hardware resources. The precise mix-and-match of horizontal and vertical clustering requires testing in your environment. This topology ensures the best possible
use of computers while providing maximum failover and load balancing. Figure 3 shows a mixed
cluster.

**Figure 3. Mixed cluster**

![Mixed cluster diagram]

**Clustering J2SE MDM CE services**

MDM CE comprises seven services that together provide the entire range of MDM functions. Each
service appears as a distinct operating system level process. The seven services are:

- Admin service
- Rmiregistry service
- Eventprocessor service
- Scheduler service
- Queuemanager service
- Workflowengine service
- Appsvr service

See [Resources](#) for more detail about these services.

The appsvr service is the J2EE application server discussed in the previous section. The rest of
the services are J2SE services (called J2SE MDM CE services from now on) and spawn a new
stand-alone JVM instance when started. The admin service and rmiregistry services must be run
on each physical system. Thus, load sharing can be implemented by instantiating one or more
of each of the remaining four services on the same or separate physical machines (vertical or
horizontal, respectively). The exception is the workflow service, which must be instantiated only
once in the entire cluster.

The J2SE MDM CE services must be instantiated at least once. However, the services that
perform most of the tasks are the MDM appsvr and the scheduler services. As a result, in a typical
scenario, there are fewer instances of eventprocessor and queuemanager services. In this tutorial,
we've instantiated at least two instances for each service (with the exception of workflow) to depict
high availability of each of the MDM CE services.

The instructional steps for clustering the J2SE MDM CE services are covered as a part of the
overall MDM cluster install.
Topology in this tutorial

This tutorial explores a mixed topology, shown in Figure 4, where the MDM CE application would be deployed on two Linux servers. The servers are typically placed within the secured firewall of an enterprise. The front end of the setup consists of two HTTP servers acting as a caching proxy and load balancer to the MDM CE application server. HTTP servers are typically placed in a demilitarized zone (DMZ) to add an additional layer of security to the local area network (LAN), providing the first line of defense against intruders. The installation of the MDM CE cluster is agnostic of such a network configuration and hence can be optional based on the organization’s network security assessment. The IP sprayer is typically a hardware device that routes requests based on the algorithms, which could be round robin or intelligent based on the load information of IHS. (Discussion of the IP sprayer is outside the scope of this tutorial.)

Figure 4. Topology

Software configurations in the topology

The topology in Figure 5 shows the MDM CE cluster (WebSphereCluster01) consisting of two physical servers (MDM SERVER 01, MDM SERVER 02), which are hosting two nodes with node agents (NODE AGENT 01, NODE AGENT 02) and a Deployment Manager (DEPLOYMENT MGR) process. Across the two nodes, we create the clusters consisting of:

- Two application servers (APPSERVER 01, APPSERVER 02) vertically clustered.
- One application server (APPSERVER03) horizontally clustered.
- J2SE MDM CE services distributed between the two servers to provide high availability and load balancing of MDM CE services.

HTTP servers (IBM HTTP SERVER 01, IBM HTTP SERVER02) act as caching proxies and balance the load of the requests.
Figure 5. Cluster components in mixed topology

Various OS level processes
Figure 6 shows various OS processes for our configuration.

Figure 6. OS process

Deploying the MDM CE clustered environment
This section provides step-by-step instructions for each of the following major deployment activities.

- **Configuring WebSphere Application Server** on Host01
  - Creating Deployment Manager (Dmgr01Host01) and a federated application server profile (AppSrv01Host01)
- **Configuring NFS for MDM CE** install directory
- **Configuring IBM IM for MDM CE Installation**
- **Installing MDM CE on AppSrv01Host01 using IM**
- **Configuring WebSphere Application Server on Host02**
  - Creating application server profile (AppSrv02Host02)
  - Federating Appsrv02Host02 to Dmgr01Host01
• Propagating MDM CE installation to mixed topology (one horizontal and one vertical)
• Configuring JVM Parameters and MDM CE parameters for the clustered members
• Adding host aliases to virtual host (MDMCE_VHOST01_HOST01)
• Installing single HTTP server and configuring it to work as a front end to cluster

Configuring WebSphere Application Server on Host01

This section describes how to create the Deployment Manager and a federated application server profile.

1. Start the Profile Management Tool (PMT) to create the profile. You can use one of the following methods to start the PMT:
   • Issue the command to open the WebSphere Customization Toolbox directly from a command prompt, then open the PMT.
   • Use the Linux window manager start menus to start the WebSphere Customization Toolbox then open the PMT.
2. In the Profile Management Tool window, click Create... on the Profiles tab, as in Figure 7, to create a new profile. The Profiles tab contains a list of profiles that have been created on your machine.
   The Environment Selection window should open.

   Figure 7. WebSphere Customization Toolbox

3. In the Environment Selection window, select Cell (deployment manager and a federated application server), then click Next, as in Figure 8.
Figure 8. PMT - cell deployment selection

4. From the Profile Creation Options window, shown in Figure 9, select **Advanced profile creation** then click **Next**.
Figure 9. PTM - Advanced profile creation

5. Select **Deploy the administrative console** from the Optional Application Deployment window, as in Figure 10, then click **Next**.
6. In the Profile Name and Location window, shown in Figure 11, provide the names and home directories for the profiles to be created. The profile names for this tutorial are as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Information to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment manager profile</td>
<td>Dmgr01Host01</td>
</tr>
<tr>
<td>Application server profile</td>
<td>AppSrv01Host01</td>
</tr>
<tr>
<td>Profile directory</td>
<td>/opt/IBM/PIM/Websphere85/AppServer/profiles</td>
</tr>
</tbody>
</table>
7. In the Node, Host, and Cell Names window in Figure 12, provide the names shown here and then click **Next**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Information to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment manager node name</td>
<td>CellManager01Host01</td>
</tr>
<tr>
<td>Application server node name</td>
<td>Node01Host01</td>
</tr>
<tr>
<td>Host name</td>
<td>Islrpeixv243.in.ibm.com</td>
</tr>
<tr>
<td>Cell name</td>
<td>Cell01Host01</td>
</tr>
</tbody>
</table>
8. In the Administrative Security window, enable administrative security by providing a user name and a password, as in Figure 13, then click **Next**.
9. In the next few windows, use the defaults until you reach the Profile Creation Summary window shown in Figure 14. Review the details, and then click **Create**.
10. After a successful profile creation, click **Finish** on the Profile Creation Complete window, as in Figure 15, to exit the wizard.
11. The PMT shows the profiles created, as shown in Figure 16.
12. Start Dmgr01Host01 manager and AppSrv01Host01 node agent from the command line, as in Figure 17, with the following commands:
   - Start Dmgr01Host01 manager
     
     ```bash
     /opt/IBM/PIM/Websphere85/AppServer/profiles/Dmgr01Host01/bin/startManager.sh
     ```
   - Start AppSrv01Host01 node agent
     
     ```bash
     /opt/IBM/PIM/Websphere85/AppServer/profiles/AppSrv01Host01/bin/startNode.sh
     ```
You can also create these profiles from the command line prompt, shown in Listing 1 and Listing 2.

**Listing 1. Dmgr01Host01 profile**

```bash
/opt/IBM/PIM/Websphere85/AppServer/bin/manageprofiles.sh -create -profileName Dmgr01Host01 -profilePath /opt/IBM/PIM/Websphere85/AppServer/profiles/Dmgr01Host01 -templatePath /opt/IBM/PIM/Websphere85/AppServer/profileTemplates/cell/dmgr -nodeProfilePath /opt/IBM/PIM/WAS8552ND/WebSphere/AppServer/profiles/Appsrv01Host01
-hostName Islrpbeixv243.in.ibm.com -nodeName CellManager01Host01 -appServerNodeName Node01Host01 -cellName Cell01Host01 -enableAdminSecurity true -adminUserName was4mdmcs -adminPassword was4mdmcs
```

**Listing 2. Appsrv01Host01 profile**

```bash
/opt/IBM/PIM/Websphere85/AppServer/bin/manageprofiles.sh -create -profileName Appsrv01Host01 -profilePath /opt/IBM/PIM/Websphere85/AppServer/profiles/Appsrv01Host01 -templatePath /opt/IBM/PIM/Websphere85/AppServer/profileTemplates/cell/default
-hostName Islrpbeixv243.in.ibm.com -nodeName CellManager01Host01 -cellName Cell01Host01 -appServerNodeName Node01Host01 -dmgrProfilePath /opt/IBM/PIM/Websphere85/AppServer/profiles/Dmgr01Host01 -enableAdminSecurity true -adminUserName was4mdmcs -adminPassword was4mdmcs
```

**Set up MDM installation directory over NFS**

MDM CE clustering requires sharing the install directory ($TOP) across its nodes. There are multiple ways to achieve this file system sharing, but the easiest is to enable the NFS server and share the location.
It is recommended that the physical location $TOP be on the database server and this location be shared across the nodes of the cluster, as in Figure 18.

**Figure 18. Typical NFS setup for MDM CE $TOP sharing**

![Diagram of typical NFS setup for MDM CE $TOP sharing](image)

In this tutorial, the database resides on Host01 so you would set up the NFS server on Host01.

1. Create the directory where MDM CE is physically stored. This tutorial assumes the location:
   
   /opt/IBM/MDMCEHOME

   Note that the above directory is not the install directory ($TOP) but merely the physical location holding MDM CE files.

2. After the directory is created, you should verify that the NFS services are running. From the command line prompt run the following commands.
   
   `/sbin/service nfs status`

   If the services are running, you will see output similar to Figure 19.

   **Figure 19. Checking NFS status using command line interface**

   ![Command output](image)

   If the services are not running, start them using the following command.

   `/sbin/service nfs start`

3. After the NFS service is running (and we assume that the firewall settings have been configured), you need to specify which part of the file system is to be accessed, or shared, by the nodes. Edit or create file `/etc/exports` to export directories for access by remote systems via NFS. The export format is:

   `<export dir>  <host1>(<options>)
<host2>(<options>)...`

4. Export the directory so that it is available for other nodes, as follows.
5. Once configured, the directories must first be exported using the `exportfs` command below.

```
/export/sbin/exportfs -a
```

Once exported, this location would be mounted by each of the cluster nodes that would be referenced in all the MDM CE configuration files. Hence, the cluster node location where the remote folder is mounted would be the MDM CE home directory ($TOP).

6. To mount on Host01, login to Host01 and from a terminal window create a folder where the remote folder is to be mounted:

```
mkdir /opt/IBM/MDMCE
```

7. Enter the command to mount the remote folder:

```
mount -v -o 'vers=3' ISLRPBEIXV243:/opt/IBM/MDMCEHOME /opt/IBM/MDMCE
```

Repeat the same process from Host02 to mount the remote folder on:

```
/opt/IBM/MDMCE
```

Assuming no errors are reported, you should be able to add or modify files from Host01 and Host02 and see changes across the nodes.

**Configuring IBM IM for MDM CE installation**

1. Start IBM IM using the following command in Figure 20.

   **Figure 20. Starting IBM IM using command line interface**

   ```
   was4pim@islrpbeixv243:~/.IBM/InstallationManager/eclipse(Tue Jul 02 - 23:18:48) $>pwd
   /home/was4pim/.IBM/InstallationManager/eclipse
   was4pim@islrpbeixv243:~/.IBM/InstallationManager/eclipse(Tue Jul 02 - 23:18:50) $>./IBMIM
   ```

   The IBM IM window should open, as in Figure 21.
Figure 21. IBM IM splash screen

2. Select **Preferences** from the IBM IM File menu, as in Figure 22.
3. From the Repositories window, as shown in Figure 23, click **Repositories** then click **Add Repository...**
4. From the Add a repository window, provide the path to the MDM CE installable and click OK, as in Figure 24.

**Figure 24. Browse to add Repository**
5. In the Repositories pane, click **Test Connections** to make sure that IBM IM is configured correctly before MDM CE deployment and click **OK** on both the windows to come back to the IBM IM main menu.

**Figure 25. Test repositories**

![Test repositories](image)

---

### Installing MDM CE on AppSrv01Host01 using IBM IM

The following are prerequisites for installing MDM CE on AppSrv01Host01 using IBM IM.

- The IBM Installation Manager V 1.6 is already configured with the MDM CE installable.
- The required environment variables have been exported before starting MDM CE installation. You can do this by inserting the following lines in ~/.bashrc, ~/bash_profile or ~/.profile depending on the default shell used.

```
source /home/db2inst1/sqllib/db2profile

# Language configuration
LANG=en_US
LC_ALL=en_US
export LANG LC_ALL

# PIM configuration
export TOP=/opt/IBM/MDMCE
export PERL5LIB=$TOP/bin/perllib
export JAVA_HOME=/opt/IBM/PIM/Websphere85/AppServer/java
export COMPAT=$TOP/bin/compat.sh
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH:
```
To begin the installation:

1. Start IBM IM and click **Install** on the main menu, as in Figure 26.

   **Figure 26. IBM IM - Install**

2. Select **IBM InfoSphere Master Data Management Collaborative Edition** and **Version 11.0.0** in the Install Packages window then click **Next**, as in Figure 27.
3. Accept the MDM CE license agreement and click **Next**, as in Figure 28.
4. In the Install Packages window shown in Figure 29, select `/opt/IBM/MDMCE` as the Installation Directory (configured earlier) and click **Next**.
5. In the Install Packages - Select the Translations to Install window, select the IBM IM display language and click **Next**, as in Figure 30.

**Figure 30. IBM IM - Select Language**

6. There are two available modes of MDM CE deployments.
• Install with IBM WAS ND: Choose this option if you need IBM IM to configure MDM CE after installation.
• Extract the product files: Choose this option if you want to configure MDM CE manually.

In the Install Packages - Select the Features to Install window, check the box to **Install with IBM WebSphere Application Server Network Deployment** and click **Next**, as in Figure 31.

**Figure 31. IBM IM - Install with first option**

7. In the Database Configuration pane, enter the values below for the database parameters then click **Test Connection** to confirm that the test connection is successful, as in Figure 32. Click **Next**.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database type</td>
<td>DB2</td>
</tr>
<tr>
<td>Database host name</td>
<td>Islrpbxixv243.in.ibm.com</td>
</tr>
<tr>
<td>Database port</td>
<td>50000</td>
</tr>
<tr>
<td>Database user name</td>
<td>db2inst1</td>
</tr>
<tr>
<td>Database password</td>
<td>*****</td>
</tr>
<tr>
<td>Local database name</td>
<td>MDMCDB</td>
</tr>
<tr>
<td>Remote database name</td>
<td>MDMCDB</td>
</tr>
<tr>
<td>Database home</td>
<td>/home/db2inst1</td>
</tr>
<tr>
<td>Database schema</td>
<td>DB2INST1</td>
</tr>
</tbody>
</table>

Figure 32. IBM IM - Configuration details for Database related parameters

8. In the WebSphere Application Server window, enter the details of the parameters listed below then click **Retrieve Host Details**, as in Figure 33. Select the **Federated** option for deployment on IBM WAS ND type.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server home</td>
<td>/opt/IBM/PIM/Websphere85/AppServer</td>
</tr>
<tr>
<td>WebSphere Application Server profile home</td>
<td>/opt/IBM/PIM/Websphere85/AppServer/profiles/Dmgr01Host01</td>
</tr>
<tr>
<td>Host name</td>
<td>Islrpbeixv243.in.ibm.com</td>
</tr>
<tr>
<td>SOAP port</td>
<td>8958</td>
</tr>
<tr>
<td>User name</td>
<td>was4mdmcs</td>
</tr>
<tr>
<td>Password</td>
<td>was4mdmcs</td>
</tr>
<tr>
<td>Cell</td>
<td>Cell01Host01</td>
</tr>
<tr>
<td>Node</td>
<td>Node01Host01</td>
</tr>
<tr>
<td>Server</td>
<td>MDMCE_APPSERVER01_HOST01</td>
</tr>
<tr>
<td>Virtual host name</td>
<td>MDMCE_VHOST01_HOST01</td>
</tr>
</tbody>
</table>

**Figure 33. IBM IM - Configuration details for Appserver related parameters**

9. In the Application Configurations window, enter the following parameters for MDM CE Application Configuration, as in Figure 34.
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERL Directory</td>
<td>/home/was4pim/ActivePerl512</td>
</tr>
<tr>
<td>JDK Path</td>
<td>/opt/IBM/PIM/Websphere85/AppServer/java</td>
</tr>
<tr>
<td>Locale</td>
<td>English</td>
</tr>
<tr>
<td>Cache Multicast Address</td>
<td>239.1.1.1</td>
</tr>
<tr>
<td>Cache Multicast TTL</td>
<td>1</td>
</tr>
<tr>
<td>RMI</td>
<td>17507</td>
</tr>
<tr>
<td>Application Server HTTP port</td>
<td>7507</td>
</tr>
</tbody>
</table>

**Note:** Multicast Address should always be 239.XX.XX.XX and it should be unique across an MDM cluster. The dev cluster cannot share the same multicast address as the QA cluster if they are on the same network.

Check the box to **Create database tables to be used by the product** then click **Next**.

**Figure 34. IBM IM - Configuration details for Application related parameters**

10. In the Summary pane, review the MDM CE parameters in the summary page and click **Next**, as in Figure 35.
Figure 35. IBM IM - Configuration Review

11. In the Install Packages - Review the Summary Information window, click **Install** to start the MDM CE installation, as in Figure 36.
12. After successful MDM CE deployment, click **Finish** in the Install Packages window, as in Figure 37.
13. Go to the $TOP/bin/go directory and run the rmi_status.sh script to confirm that all the MDM CE services are running, as in Figure 38.

**Figure 38. rmi_status.sh execution using command line interface**

```
was4pim@islrpeixv243:~/IBM/MDMCE/bin/go(Wed Jul 03 - 00:20:13)
$>rmi_status.sh
++ [success] rmistatus (Wed Jul 03 00:20:19 IST 2013)
//islrpeixv243:17507/db2inst1/eventprocessor/eventprocessor_ISLRPBEIXV243
//islrpeixv243:17507/db2inst1/scheduler/scheduler_ISLRPBEIXV243
//islrpeixv243:17507/db2inst1/queuemanager/queuemanager_ISLRPBEIXV243
//islrpeixv243:17507/db2inst1/workflowengine/workflowengine_ISLRPBEIXV243
//islrpeixv243:17507/db2inst1/admin/admin_ISLRPBEIXV243
//islrpeixv243:17507/db2inst1/appsvr/appsvr_ISLRPBEIXV243
was4pim@islrpeixv243:~/IBM/MDMCE/bin/go(Wed Jul 03 - 00:20:19)
$>
```

14. Run the $TOP/bin/go/stop_local.sh script to stop all the MDM CE services.

At this point, MDM CE is installed in standalone mode on a single application server.

**Configuring WebSphere Application Server on Host02**

This section describes how to create a custom node and federate it to the Deployment Manager.

1. Start the Profile Management Tool (PMT) to create the profile.
2. From the Profiles tab, click Create... to create a new profile, as in Figure 39.

**Figure 39. PMT - Create new profile**

3. From the Environment Selection window, select **Custom profile** then click **Next**, as in Figure 40.
4. In the Profile Creation Options window shown in Figure 41, select **Advanced profile creation** then click **Next**.
5. In the Profile Name and Location window, provide the name and directory of the custom profile to be created using the information below. Then click **Next**, as in Figure 42.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile name</td>
<td>AppSrv02Host02</td>
</tr>
<tr>
<td>Profile directory</td>
<td>/opt/IBM/PIM/Websphere85/AppServer/profiles/AppSrv02Host02</td>
</tr>
</tbody>
</table>
6. In the Node and Host Names window, specify the node and hostname for the custom profile using the information below then click **Next**, as in Figure 43.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node name</td>
<td>Node02Host02</td>
</tr>
<tr>
<td>Host name</td>
<td>Islrpbeixv373.in.ibm.com</td>
</tr>
</tbody>
</table>
7. Specify the details of the existing Deployment manager for federation, such as deployment manager host name and SOAP port and security credentials, using the information below.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment manager host name</td>
<td>Islrpbeixv243.in.ibm.com</td>
</tr>
<tr>
<td>Deployment manager SOAP port</td>
<td>8958</td>
</tr>
<tr>
<td>Username</td>
<td>was4mdmcs</td>
</tr>
<tr>
<td>Password</td>
<td>*****</td>
</tr>
</tbody>
</table>

8. Uncheck the box for **Federate this node later** and click **Next**, as in Figure 44.
9. Click **Next** on the subsequent windows until you see the Profile Creation Summary window, as in Figure 45. Verify that the details entered are valid then click **Create**.
10. After successful creation of the profile, the PMT will show the custom profile that you created, as in Figure 46.
11. Start the AppSrv02Host02 node agent the using following command.

```
/opt/IBM/PIM/Websphere85/AppServer/profiles/AppSrv02Host02/bin/startNode.sh
```

12. **Note:** You can also create the above-mentioned custom profile from the command prompt, as follows.

**Listing 3. Appsrv02Host02 profile**

```
/opt/IBM/PIM/Websphere85/AppServer/bin/manageprofiles.sh -create -profileName Appsrv02Host02 -profilePath /opt/IBM/PIM/Websphere85/AppServer/profiles/Appsrv02Host02 -templatePath /opt/IBM/PIM/Websphere85/AppServer/profileTemplates/managed -hostName Islrpbeixv243.in.ibm.com -nodeName Node02Host02 -federateLater false -dmgrHost Islrpbeixv243.in.ibm.com -dmgrPort 8953 -enableAdminSecurity true -adminUserName was4mdmcs -adminPassword was4mdmcs
```

**Creating a cluster of mixed topology**

This section shows how to span the existing MDM CE application onto a WebSphere cluster.

1. Login to WebSphere administrative console on Host01 and open the WebSphere application server clusters window by navigating to **Servers->Clusters->WebSphere application server clusters** in the left pane, as in Figure 47.

   Click **New** to create a cluster with the cluster name **MDMCEClusterHost01**.
2. In the Create a New Cluster window, clear the box to **Configure HTTP session memory-to-memory replication** and click **Next**, as in Figure 48.

**Figure 48. Create a new Cluster - Cluster name**

3. In the Create First Cluster Member pane, convert the existing application server **MDMCE_APPSERVER01_HOST01** with a Weight of 2 as the first member of the cluster **MDMCEClusterHost01**, as in Figure 49, then click **Next**.
4. In the Create Additional Cluster Members pane shown in Figure 50, enter the second Member name MDMCE_APPSERVER02_HOST01 with a Weight of 2 on node Node01Host01.

5. Click Add Member.

Figure 50. Create a new Cluster - Second member details (vertical)

Now the MDMCE_APPSERVER02_HOST01 server is vertically clustered to MDMCE_APPSERVER01_HOST01 server.

Note: Select the Generate unique HTTP ports option because MDMCE_APPSERVER02_HOST01 is also present on HOST01(same physical machine as MDMCE_APPSERVER01_HOST01). In general, the Generate unique HTTP ports option is selected while adding vertical cluster members.
6. Enter the third Member name of cluster as MDMCE_APPSERVER03_HOST02 with a Weight of 2 on node Node02Host02 then click Add Member, as in Figure 51.

**Figure 51. Create a new Cluster - Third member details (horizontal)**

![Create a new Cluster - Third member details (horizontal)](image)

Now the MDMCE_APPSERVER03_HOST02 server is horizontally clustered to MDMCE_APPSERVER01_HOST01 server.

**Note:** The Generate unique HTTP ports option is not selected since MDMCE_APPSERVER03_HOST02 is present on HOST02 (a physically different machine).

7. As shown in Figure 52, now all three members of the cluster MDMCEClusterHost01 are visible at the bottom of the Create Additional Cluster Members pane. Click Next.
8. In the Summary pane, click **Finish** to create the cluster, as in Figure 53.

**Figure 53. Create a new Cluster - Summary**
9. From the Synchronize Changes with Nodes window shown in Figure 54, review the changes and synchronize the changes with all the nodes of the cluster.

**Figure 54. Create a new Cluster - Sync the changes**

10. To review the cluster created, click **Servers->Clusters->Cluster topology** and expand the MDMCEClusterHost01 in the Cluster Topology window, as in Figure 55.

**Figure 55. Cluster Topology**

11. Click **Servers->Clusters->WebSphere application server clusters**. The cluster that you created is in the left pane, as in Figure 56.
12. Make a note of the HTTP ports of the cluster MDMCEClusterHost01 members (MDMCE_APPS01_HOST01, MDMCE_APPS02_HOST01, MDMCE_APPS03_HOST02 application servers).

- For MDMCE_APPS01_HOST01 the port is 7507, which you can obtain by traversing to Servers->WebSphere application servers->MDMCE_APPS01_HOST01->Ports and search for WPC port.
- For MDMCE_APPS02_HOST01 the port is 7515, which you can obtain by traversing to Servers->WebSphere application servers->MDMCE_APPS02_HOST01->Ports and search for WPC port.
- For MDMCE_APPS03_HOST02 the port is 7515, which you can obtain by traversing to Servers->WebSphere application servers->MDMCE_APPS03_HOST02->Ports and search for WPC port.

Configuring JVM parameters and MDM CE parameters for the clustered members

Go to Servers->WebSphere Application servers->MDMCE_APPS02_HOST01->Process definition->Java Virtual Machine and modify appsrv to appsrv1 in the generic JVM arguments of the MDMCE_APPS02_HOST01 application server for the following parameters.

-Dsvc_name=appsvr1_ISLRPBEIXV243
-Dsysout.dir=/opt/IBM/MDMCE/logs/appsvr1_ISLRPBEIXV243

Post modifications, the generic JVM arguments will be:

-Dsvc_name=appsvr1_ISLRPBEIXV243
-Dsysout.dir=/opt/IBM/MDMCE/logs/appsvr1_ISLRPBEIXV243
-DTOP=/opt/IBM/MDMCE
-DCCD_ETC_DIR=/opt/IBM/MDMCE/etc
-Dsvc etc_dir=/opt/IBM/MDMCE/etc/default
-Dtrigo.memflags=-Xmx1024m_-Xms256m
-Djava.security.policy=/opt/IBM/MDMCE/etc/default/java.policy
-Dexit_if_config_file_not_found=false
-DenableJava2Security=true
-Dsysout.dir=/opt/IBM/MDMCE/logs/appsvr1_ISLRPBEIXV243
Go to **Servers** -> **WebSphere Application servers** -> **MDMCE_APPSERVER03_HOST02** -> **Process definition** -> **Java Virtual Machine** and change the host name. For example, change `ISLRPBEIXV243` to `ISLRPBEIXV373` in the generic JVM arguments of the MDMCE_APPSERVER02_HOST01 application server for the following parameters.

```
-Dsvc_name=appsvr1_ISLRPBEIXV373
-Dsysout.dir=/opt/IBM/MDMCE/logs/appsvr1_ISLRPBEIXV373
```

Now the generic JVM arguments will be:

```
-Dsvc_name=appsvr1_ISLRPBEIXV373
-DTOP=/opt/IBM/MDMCE
-DCCD_ETC_DIR=/opt/IBM/MDMCE/etc
-Dsvc_etc_dir=/opt/IBM/MDMCE/etc/default
-Dtrigo.memflags=-Xmx1024m_-Xms256m
-Djava.security.policy=/opt/IBM/MDMCE/etc/default/java.policy
-Dexit_if_config_file_not_found=false
-DenableJava2Security=true
-Dsysout.dir=/opt/IBM/MDMCE/logs/appsvr1_ISLRPBEIXV373
```

### Configuring MDM CE parameters for the clustered members

You need to make MDM services aware of the newly created cluster and its nodes. Edit the `$TOP/etc/default/admin_properties.xml` to add two host names, as follows.

```
<admin>
  <cluster>
    <host name="islrpbeixv243"/>
    <host name="islrpbeixv373"/>
  </cluster>
</admin>
```

At this point, the two nodes are sharing the same MDM configuration files. Because the service list needs to be different at both the nodes, you need to provide two different configuration files for both the nodes. To achieve this you can use the following steps.

1. Create a folder named cluster under the `$TOP` directory.
2. Create folders mdmapp1 and mdmapp2 within the cluster folder you just created. These will hold the configuration files for horizontal nodes.
3. Copy the content of `$TOP/bin/conf` to both mdmapp1 and mdmapp2.
4. Export `CCD_CONFIG_DIR=/opt/IBM/MDMCE/cluster/mdmapp1` (for host1.)
5. Export `CCD_CONFIG_DIR=/opt/IBM/MDMCE/cluster/mdmapp2` (for host2).

For Host02, edit relevant sections in `$TOP/cluster/ mdmapp2/env_settings.ini` to reflect the application server profile, node, and appserver namer/mdmapp2 (for host2), as follows.

```
[appserver.websphere]
admin_security=true
application_server_profile=AppSrv02Host02
cell_name=Cell01Host01
node_name=Node02Host02

[appserver.appsvr]
port=7507
appserver_name=MDMCE_APPSERVER03_HOST02
vhost_name=MDMCE_VHOST01_HOST01
```
For Host01, edit relevant sections in $TOP/cluster/mdmapp1/env_settings.ini to reflect the additional application server with its detail. Also add the server to the list of services under the “services” section.

```
[appserver.appsvr1]
port=7508
appserver_name=MDMCE_APPSERVER02_HOST01
vhost_name=MDMCE_VHOST01_HOST01
```

Under services sections add appsrv1.

```
[services]
admin=admin
queue.manager=queue.manager
event.processor=event.processor
scheduler=scheduler,scheduler01
workflow.engine=workflow.engine
appsvr=appsvr
```

**Adding more J2SE MDM CE services**

As discussed in [Clustering J2SE MDM CE Services](#), J2SE MDM CE services can be instantiated multiple times to load balance the services. The most likely service that is load balanced is the scheduler. The workflow engine cannot be instantiated more than once, and eventprocessor and queue.manager are the least likely services in a typical implementation and thus can have single instances or a maximum of two instances to provide high availability.

As shown in [Figure 6](#), you need to configure two instances of the scheduler. The default configurations usually have at least one service configured so you would need to add another instance of the scheduler.

For Host02, go to the services section and add an additional scheduler service. The name of the service to add is scheduler01.

```
[services]
admin=admin
queue.manager=queue.manager
event.processor=event.processor
scheduler=scheduler,scheduler01
workflow.engine=workflow.engine
appsvr=appsvr
```

**Adding host aliases to virtual host**

On the IBM WebSphere administrative console, go to **Environment->Virtual Hosts -> MDMCE_VHOST01_HOST01->Host Aliases** and add the HTTP ports of the other two cluster members. After adding these ports to MDMCE_VHOST01_HOST01 you should see something similar to the Host Aliases window in [Figure 57](#).
IBM HTTP Server

This section explains how to install and configure the IBM HTTP Server and Web Server Plug-in. IBM HTTP server sprays web requests to the cluster members for balancing the work load among relevant application servers. You can specify the strategy for load balancing and the necessary parameters in the plugin-cfg.xml file. The default, and most commonly used strategy for workload balancing, is weighted round robin. For details refer to the IBM Redbooks® technote, Workload Management Policies. For the sake of brevity, in this tutorial only one HTTP server configuration is described but you can easily follow the steps to configure another instance of the HTTP server which, in this tutorial, is load balanced by a hardware load balancer.

**Note:** Technically, HTTP Server is an optional component but is recommended in a production environment.

### Installing IBM HTTP Server and Plug-in

Follow the steps below to install the IBM HTTP Server and Plug-in.

1. Start IBM IM and configure WebSphere Application Server supplements installable to the IBM IM repository.
2. Select **Install** on the IBM IM main menu, as in Figure 58.
3. In the Install Packages window, select **IBM HTTP Server** and **Web Server Plug-in** for IBM WebSphere Application Server and their respective versions. Click **Next**, as in Figure 59.
4. As shown in Figure 60, accept the license agreement for IBM WebSphere Application Server and click **Next**.
5. Enter IBM HTTP Server and Web Server Plug-in destination directories:
   • /opt/IBM/PIM/WebSphere85/HTTPServer
   • /opt/IBM/PIM/WebSphere85/Plugins

   as in Figure 61 and click **Next**.
6. Specify the IBM HTTP Server port number 8080 and click **Next**, as in Figure 62.
Figure 62. IBM IM - Choose HTTP port

7. Click **Install** on the final Install Packages window that lists all the packages, as in Figure 63.
8. Click **Finish** after getting the message for successful installation of IBM HTTP Server.
Figure 64. IBM IM - Finish the installation

Creating a web server instance through WebSphere Application Server console

- Open the IBM WebSphere Application Server console using the secure port with the following credentials:
  - Username: was4mdmcs
  - Password: was4mdmcs
- Click **Servers -> Server Types -> Web Servers** in the left pane of the console, as in Figure 65.
• Click **New** to create a new Web server. In the Select a node for the Web server and Select the Web Server Type pane shown in Figure 66, select node **Node02Host02**, Server name **IBMHTTPServer01**, and Type **IBM HTTP Server**, then click **Next**.

**Figure 66. Web Server Creation - Enter name and node**

• In the Select a Web Server Template pane, select **IHS** and click **Next**, as in Figure 67.
Figure 67. Web Server Creation - Select Web Server Template

![Web Server Creation - Select Web Server Template](image)

- Enter the properties of the new Web server created using the information below. Click **Next**, as in Figure 68.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>8080</td>
</tr>
<tr>
<td>Web server installation location</td>
<td>/opt/IBM/PIM/Websphere85/HTTPServer</td>
</tr>
<tr>
<td>Plug-in installation location</td>
<td>/opt/IBM/PIM/Websphere85/Plugins</td>
</tr>
<tr>
<td>Application mapping to the Web server</td>
<td>All</td>
</tr>
</tbody>
</table>

The default port for HTTP is port 80. If you do not intend to front end the web server with an additional load balancer, the preference for port would be 80 (which may need additional system level permissions). For this tutorial, the web servers are going to be front ended by the hardware load balancer so we choose port 8080.

**Figure 68. Web Server Creation - Enter Properties**

![Web Server Creation - Enter Properties](image)

- Click **Finish** in the Confirm new Web Server pane to confirm the creation of the new web server, as in Figure 69.
Figure 69. Web Server Creation - Review summary

Configuring IHS for Application Server

The plug-in enables IBM HTTP Server (IHS) to communicate with the cluster. To use this plug-in, you must enable the HTTP Server with the details of the plug-in. The configuration changes are typically recorded in the httpd.conf file of IHS.

1. Run the ConfigureIHSPlugin.sh script to configure the IHS to use the Web server plug-in. The command is in the /opt/IBM/PIM/Websphere85/Plugins/bin directory.

   ConfigureIHSPlugin.sh -plugin.home /opt/IBM/PIM/Websphere85/Plugins
   -plugin.config.xml /opt/IBM/PIM/Websphere85/Plugins/config/IBMHTTPServer01/plugin-cfg.xml
   -ihs.conf.file /opt/IBM/PIM/Websphere85/HTTPServer/conf/httpd.conf
   -ihs.admin.usergroup was4pim
   -operating.system Linux -operating.system.arch 64
   -WAS.webserver.name IBMHTTPSERVER01 WAS.host.name islrpbe1xv373.in.ibm.com

2. Go to Servers->Server Types->Web Servers and select IBMHTTPServer01. Click Generate Plug-in, shown in Figure 70.
3. Select the Web Server IBMHTTPServer01 and click Propagate Plug-in to synchronize the generated plug-in, as in Figure 71.

**Figure 71. Web Server Console - Propagate Plug-in**

The plugin-cfg.xml file gets generated under /opt/IBM/PIM/Websphere85/Plugins/config/IBMHTTPServer01.

4. In the Host Aliases window, add Web server port 8080 to host aliases of the virtual Host (MDMCE_VHOST01_HOST01), as in Figure 72.
Starting the cluster

To start the cluster:

1. Start Deployment Manager and node agents.
2. Log in to Host2 and issue the command `$TOP/bin/go/start_local.sh`.
3. Log in to Host1 and issue the command `$TOP/bin/go/start_local.sh`.
4. At this point, all the services including application servers should have started. To verify the status, issue the command `$TOP/bin/go/rmi_status.sh`, which should bring up the list of services running in the entire MDM cluster, as in Figure 73.

**Figure 73. rmi_status.sh using command line interface**

```
$TOP/bin/go/rmi_status.sh
++ [success] rmistatus (Sun Sep 08 11:02:21 IST 2013)
//isrpeixv243:17507/db2inst1/scheduler/scheduler_ISLRPBEIXV243
//isrpeixv243:17507/db2inst1/queuemanager/queuemanager_ISLRPBEIXV243
//isrpeixv243:17507/db2inst1/admin/admin_ISLRPBEIXV243
//isrpeixv243:17507/db2inst1/appsvr/appsvr_ISLRPBEIXV243
//isrpeixv243:17507/db2inst1/appsvr1/ISLRPBEIXV243
//isrpeixv373:17507/db2inst1/apps/controlmanager/ISLRPBEIXV373
//isrpeixv373:17507/db2inst1/admin/admin_ISLRPBEIXV373
//isrpeixv373:17507/db2inst1/appsvr/appsvr1_ISLRPBEIXV373
//isrpeixv373:17507/db2inst1/eventprocessor/eventprocessor_ISLRPBEIXV373
//isrpeixv373:17507/db2inst1/scheduler/scheduler01_ISLRPBEIXV373
//isrpeixv373:17507/db2inst1/scheduler01/ISLRPBEIXV373
//isrpeixv373:17507/db2inst1/workflowengine/workflowengine_ISLRPBEIXV373
```

5. Go to the WebSphere Application Server console, navigate to Servers->Server Types-> Web servers and select IBMHTTPServer01, then click Start, as in Figure 74.

**Note:**
- The tutorial workflow engine is configured to run on Host02, hence the start_local.sh command on Host02 is issued first.
• Even though the workflow engine is configured on both the nodes, the workflow engine can only be running in one server. The second workflow engine attempts to come up but will shut down automatically.

At this point, the web server should start listening to port 8080 and start redirecting the request to one of the three application servers.

**Figure 74. Web server status**

6. To verify fail over, shut down the application servers one by one in sequence to make sure the requests are still being served.

**Conclusion**

MDM CE offers a solid basis on which to build a scalable and highly available application environment. This tutorial is a starting point for those customers and practitioners who need to set up a basic clustered MDM CE installation.

This tutorial described how to set up a mixed clustered topology that reflects the recommended configuration for production environments in IBM MDM CE V11. While this tutorial provided a simple example of such a setup, you can now extrapolate the necessary steps to create more complex or slightly different topological schemes and adapt the steps to a number of different circumstances to ensure different levels of scalability and availability.

Stay tuned for Part 2 of this series, which will cover deploying MDM CE directly on an existing cluster. It will also provide details about MDM CE fix pack upgrades and advanced topics such as setting up dynamic clusters.

**Acknowledgements**

The authors would like to thank the following people for their enormous contributions, significant reviews, and suggestions to the tutorial:

• Mark Rahmani (rahmani@us.ibm.com) – System Consultant, IM InfoSphere Infrastructure
• Ramesh Panchal (ramesh.panchal@in.ibm.com) - Systems Management Specialist: WebSphere Application Server
Resources

- Learn more about InfoSphere Master Data Management Collaboration Server.
- The IBM InfoSphere MDM Version 11.0 Information Center contains the documentation for the IBM InfoSphere Master Data Management (InfoSphere MDM) technology and features.
- Get the System requirements for InfoSphere Master Data Management including the Collaborative Edition.
- Read more about DMZ, a physical or logical subnetwork that contains and exposes an organization's external-facing services to a larger and untrusted network.
- Read the IBM Redbooks tip about IBM WebSphere Application Server Network Deployment - Workload Management Policies.
- Visit the developerWorks Information Management zone to find more resources for DB2 developers and administrators.
About the authors

Sachin Prasad

Sachin Prasad is an MDM solution architect with IBM India Software Lab Services. He has been with IBM for a decade, with various roles, and has experience in product architecture, product development, solutioning, enterprise integration, customer enablement, and technical evangelism. Sachin currently consults for pre-sales, post-sales, delivery excellence, and thought leadership on MDM Collaboration Server. He can be reached at sachin.prasad@in.ibm.com.

Abhijith Boompelly

Abhijith Boompelly is a transition and quality assurance engineer on the IBM MDM Collaboration Server team. He previously was a performance specialist on the MDM performance and benchmarks team. Abhijith currently consults in the areas of process integration, implementation, and deployment using IBM WebSphere products. He can be reached at abhijith.boompelly@in.ibm.com.

Srinath Shenoy

Srinath Shenoy has managed the IBM MDM Collaboration Server Engineering team for the last five years. Previously he worked on Informix as the lead for data migration tools, developing and maintaining open connectivity APIs for IBM Informix databases. Srinath's primary interests are data analytics and open APIs. He can be reached at shenoysri@in.ibm.com.

© Copyright IBM Corporation 2014
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)