

IBM Maximo Asset Management
Version 7 Release 6

Installation Guide
(IBM WebSphere)



Note

Before using this information and the product it supports, read the information in [“Notices” on page 61.](#)

This edition applies to version 7, release 6, modification 1, fix pack 2 of IBM® Maximo® Asset Management and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2007, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Running the Maximo Asset Management 7.6.1.2 installation program.....	1
Configuring Maximo Asset Management 7.6.1.2 using the Maximo Asset Management 7.6.1.2 configuration program.....	1
Chapter 2. Creating the NOTF JMS queue.....	3
Chapter 3. Creating the NOTFERR JMS queue.....	5
Chapter 4. Update the application database with the UpdateDB Lite command.....	7
Updating the database with UpdateDB Lite in Maximo Asset Management.....	7
Chapter 5. Maximo Asset Management installation and configuration programs log information.....	9
Chapter 6. Post installation tasks.....	11
Configuring initial data.....	11
Synchronizing users and groups.....	14
Adding sample data to the Maximo Asset Management database post configuration.....	14
Enabling best-practice links.....	16
Chapter 7. Maintaining EAR files.....	17
Building EAR files.....	17
Manually deploying Work Center WAR files.....	17
Manually uninstalling applications from WebSphere Application Server Network Deployment.....	18
Manually installing applications in WebSphere Application Server Network Deployment.....	18
Chapter 8. Command line configuration.....	21
Configuration parameter validation.....	21
Input properties files.....	22
Command line interface parameters.....	23
Command line configuration program actions.....	33
action updateDatabaseConfiguration.....	33
action validateDatabaseConfiguration.....	33
action validateAndUpdateDatabaseConfiguration.....	33
action deployDatabaseConfiguration.....	34
action removeDatabaseConfiguration.....	34
action enableDB2TextSearch.....	34
action updateJ2eeConfiguration.....	35
action validateJ2eeConfiguration.....	35
action validateAndUpdateJ2eeConfiguration.....	35
action deployJ2eeConfiguration.....	35
action removeJ2EEConfiguration.....	36
action deployConfiguration.....	36
action removeConfiguration.....	36
action reapplyAppServerTuning.....	36
action enableAppSecurity.....	36
action disableAppSecurity.....	37
action updateApplication.....	37
action addLanguages.....	37

action configureProducts.....	38
action deployMiddlewareConfiguration.....	38
action upgrade75Configuration.....	39
action updateApplicationDBLite.....	39
action configureProductsDBLite.....	40
Configuration program properties.....	40
Chapter 9. Backup and restoration.....	59
Backing up Maximo Asset Management	59
Restoring the Maximo Asset Management administrative server.....	60
Notices.....	61
Index.....	65

Chapter 1. Running the Maximo Asset Management 7.6.1.2 installation program

Use the Maximo Asset Management 7.6.1.2 installation program to install Maximo Asset Management 7.6.1.2.

Before you begin

Before you can use Maximo Asset Management 7.6.1.2 with Oracle Weblogic server, you must update the Oracle 18c database configuration. For more information, see <http://www.ibm.com/support/docview.wss?uid=ibm10881398>.

About this task

In order to install Maximo Asset Management 7.6.1.2, run the Maximo Asset Management 7.6.1.2 installation program on the Maximo Asset Management administrative system.

Procedure

1. Download the feature pack compressed file and copy it to the administrative server machine.
2. Run the Installation Manager from the following directory:

Option	Description
Windows	Start > All Programs > IBM Installation Manager > IBM Installation Manager
UNIX	In the /opt/IBM/InstallationManager/eclipse directory, run the IBMIM command.

3. In the Installation Manager main window, click **File > Preferences**.
4. Click **Add Repository**.
5. Add the repository for Maximo Asset Management 7.6.1.2, click **OK**.
6. In the Preferences dialog box, click **OK**.
7. Click the Update icon.
8. In the Update Packages dialog box, select the IBM Tivoli's process automation suite package group, and click **Next**.
9. To complete the installation, follow the on-screen instructions.
10. After the installation is complete, select the option to start the Maximo Asset Management 7.6.1.2 configuration program, and then click **Finish**.

The Maximo Asset Management 7.6.1.2 installation program exits and the Maximo Asset Management 7.6.1.2 configuration program is started automatically.

What to do next

Use the Maximo Asset Management 7.6.1.2 configuration program to configure Maximo Asset Management 7.6.1.2.

Configuring Maximo Asset Management 7.6.1.2 using the Maximo Asset Management 7.6.1.2 configuration program

Use the Maximo Asset Management 7.6.1.2 configuration program to configure Maximo Asset Management 7.6.1.2.

Procedure

1. Start the configuration program:

Option	Description
Windows 8 and above	<i>InstallLocation</i> \ConfigTool\ConfigUI.exe. For example, C:\IBM\SMP\ConfigTool\ConfigUI.exe.
UNIX	<i>InstallLocation/ConfigTool/ConfigUI</i> . For example, /opt/IBM/SMP/ConfigTool/ConfigUI.

2. In the Deployment Operations pane, select **Update Database** and **Build and Deploy Application EAR Files**.
3. To complete the configuration, follow the on-screen instructions.

Results

Maximo Asset Management 7.6.1.2 is installed and configured to use WebSphere® Application Server Network Deployment.

Chapter 2. Creating the NOTF JMS queue

You must create a JMS queue NOTF for notifications.

About this task

You must create a JMS queue called NOTF to enable notifications. If you do not use the Maximo Asset Management configuration program to automatically configure WebSphere Application Server Network Deployment, you must create this queue manually.

For Maximo Asset Management multitenancy environments, the system provider must create the NOTF queue manually to enable notifications for each tenant. These queues cannot be created automatically by using Maximo Asset Management installation and configuration programs.

Procedure

1. From the WebSphere Application Server Network Deployment administrative console, click **Resources** > **JMS** > **Queues**.
2. From the **Scope** drop-down list, select **Cell=ctgCell01**.
3. Click **New**.
4. Verify that the default messaging provider is selected and then click **OK**.
5. Enter the following information, and then click **OK**.

Name

NOTF.

This value must contain only uppercase letters.

JNDI name

jms/maximo/int/queues/notf

Bus name

intjmsbus.

Queue name

CQINBD.

6. Click **OK** and then click **Save**.

Chapter 3. Creating the NOTFERR JMS queue

After creating the Error Queue Bus Destination, you create the NOTFERR Error queue for notification error handling.

About this task

You must create a JMS queue called NOTFERR to enable notification error handling. If you do not use the Maximo Asset Management configuration program to automatically configure WebSphere Application Server Network Deployment, you must create this queue manually.

For Maximo Asset Management multitenancy environments, the system provider must create the NOTF queue manually to enable notifications for each tenant. These queues cannot be created automatically by using Maximo Asset Management installation and configuration programs.

Procedure

1. From the WebSphere Application Server Network Deployment administrative console, click **Resources** > **JMS** > **Queues**.
2. From the **Scope** drop-down list, select **Cell=ctgCell01**.
3. Click **New**.
4. Verify that the default messaging provider is selected and then click **OK**.
5. Enter the following information, and then click **OK**.

Name

Enter NOTFERR.

This value must contain only uppercase letters.

JNDI name

Enter `jms/maximo/int/queues/notferr`

Bus name

Select **intjmsbus**.

Queue name

Select **CQINERRBD**.

6. Click **OK** and then click **Save**.

Chapter 4. Update the application database with the UpdateDB Lite command

UpdateDB Lite splits the tasks of preparing class files and updating the database in Maximo Asset Management.

In Maximo Asset Management 7.6.1, you can minimize downtime by using new programs to manually perform configuration steps. Tasks that are performed by the UpdateDB Lite program, preparation of product class files and database updates, are split into two separate commands. These commands include pre-processing and post-processing.

Pre-processing tasks prepare class files before they are built into the application EAR files without stopping the application server. The updatedb command can also perform this work, but it stops the application server.

In post-processing, UpdateDB Lite can perform updates to the database, but the application server must be stopped before you run the command. UpdateDB and UpdateDB Lite use the exact same parameters.

The Maximo Asset Management 7.6.1 configuration tool uses the UpdateDB to complete configuration.

Updating the database with UpdateDB Lite

Using UpdateDB Lite to update the database can reduce application downtime.

Before you begin

Use Installation Manager to install Maximo Asset Management 7.6.1. Do not use the Maximo Asset Management configuration tool to update the database or build and deploy application EAR files.

Procedure

1. Open the config tool.
2. Click **Update Database with UpdateDB Lite Feature**.
3. Verify the application version and current state.
4. Select to run the pre-processing or post-processing options, and click **Finish**.

Results

If post-processing options are selected, the Maximo application closes until the updates to the database are complete.

Chapter 5. Installation and configuration programs log information

Log files generated by the Maximo Asset Management installation and configuration programs can provide details on completed and configuration tasks and any errors that might occur.

Maximo Asset Management installation and configuration programs log locations

Maximo Asset Management and associated middleware is installed through Installation Manager. The log files of Installation Manager are common to any installation, regardless of which components you choose to install. Log files for the Maximo Asset Management installation program are found on the system where it is run.

Log file directory
C:\ProgramData\IBM\Installation Manager\logs
/var/ibm/InstallationManager/logs
C:\IBM\SMP\ConfigTool\logs
opt/IBM/SMP/ConfigTool/logs

In addition, if you receive an error credited to the J2EE server, examine the logs in the log directories for the deployment manager, node agent, and WebSphere Application Server application server.

When engaging IBM product support services, be prepared to provide these log files in an archive file. The LogZipper utility provided in the *install_home*\configtool\scripts directory can be used for this task. If you use the LogZipper utility, all relevant log files are archived in *install_home*/configtool/debug/YYYYMMDD_hhmmss.zip.

Chapter 6. Post installation tasks

There are several post installation tasks required in order to complete the Maximo Asset Management deployment.

Configuring initial data

After you successfully install and configure Maximo Asset Management components, you must complete several data configuration tasks before you use Maximo Asset Management.

Before you begin

- Create security groups and add users to the security groups.
- If you are using a directory server as part of your deployment, ensure that the user names are created in your LDAP repository.
- To ensure that you have the accounting information that you need, consult the Financial department of your organization.

Procedure

1. Log in to Maximo Asset Management by entering the user name `maxadmin` and the password that you specified during the installation.
2. If you didn't set the SMTP server by using the Configuration tool, configure the SMTP server to send users email notifications of system events.
To learn about the range of system properties in Maximo Asset Management, see the System properties information.
 - a) From the **Go To** menu, select **System Configuration > Platform Configuration > System Properties**.
 - b) In the **Global Properties** table window, use the filter feature to search for and expand the **mail.smtp.host** property.
 - c) In the **Global Value** field, specify the SMTP host name.
 - d) Select the **mail.smtp.host** check box.
 - e) From the **Common Actions** menu, click **Live Refresh**.
 - f) In the **Live Refresh** window, click **OK**.
 - g) In the **Global Properties** table window, use the filter feature to search for the **mxe.adminEmail** property.
 - h) Expand the **mxe.adminEmail** property and in the **Global Value** field, specify your email address.
 - i) Select the **mxe.adminEmail** record check box.
 - j) From the **Common Actions** menu, click **Live Refresh**.
 - k) In the **Live Refresh** window, click **OK**.
3. Define a currency code.
 - a) From the **Go To** menu, select **Financial > Currency Codes**
 - b) Click **New Row** and specify a currency code and a description.
For example, enter USD for United States of America Dollar.
 - c) Click **Save Currency**.
4. Define item sets and company sets.
 - a) From the **Go To** menu, select **Administration > Sets**.
 - b) Click **New Row**.

- c) Specify an item set name.
For example, enter IT Items.
 - d) In the **Type** field, specify ITEM.
 - e) Click **New Row**.
 - f) Specify a company set name.
For example, enter IT Comps.
 - g) In the **Type** field, specify COMPANY.
 - h) Click **Save Sets**.
5. Create an organization.
- a) From the **Go To** menu, select **Administration > Organizations**.
 - b) From the toolbar, click **New Organization**.
 - c) Specify an organization name.
For example, enter EAGLENA, and enter a long description.
 - d) Specify the base currency that you defined in step 3.
 - e) Specify the item set and the company set that you defined in step 4.
 - f) In the **Default Item Status** field, set the status to PENDING.
 - g) Click the **Sites** tab and select **New Row** under the **Sites** table window.
 - h) Specify a site name.
For example, enter Factory01 and enter a long description.
 - i) Click **Save Organization**.
6. Create a general ledger account component.
- a) From the **Go To** menu, select **System Configuration > Platform Configuration > Database Configuration**.
 - b) From the **More Actions** menu, select **GL Account Configuration**.
 - c) Click **New Row**.
 - d) Specify a component name.
For example, enter MYCOMPONENT.
 - e) Specify a length for the component.
For example, enter 5.
 - f) Specify a type for the component.
For example, select Alphanumeric.
 - g) Click **OK**.
7. Apply changes to the database.
- a) From the **More Actions** menu, select **Manage Admin Mode**.
 - b) Select **Turn Admin Mode ON** and click **OK**.
This task takes several minutes to complete. You can click **Refresh Status** to view the progress.
 - c) From the **More Actions** menu, select **Apply Configuration Changes**.
Ensure that the status To Be Changed does not appear in the status column of the listed objects.
 - d) From the **More Actions** menu, select **Manage Admin Mode**.
 - e) Select **Turn Admin Mode OFF** and click **OK**.
If you do not turn off Admin Mode, cron tasks fail.
8. Create a general ledger account.
- a) From the **Go To** menu, select **Financial > Chart of Accounts**.
 - b) In the **Organizations** table window, select your organization.
 - c) From the **More Actions** menu, select **GL Component Maintenance**.

- d) In the **Components** table window, select the component that you entered in step 6 and click **New Row**.
 - e) Specify a GL Component value and a description, and then click **OK**.
 - f) In the **GL Accounts** table window, click **New Row**.
 - g) Specify a General Ledger Account and click **Save GL Account**.
 - h) From the **Go To** menu, select **Administration > Organizations**.
 - i) Find your organization and select the record.
 - j) In the **Clearing Account** field, specify the general ledger account that you created.
 - k) Select **Active**.
 - l) Click **Save Organization**.
9. Authorize a security group to modify a general ledger component type.
- a) From the **Go To** menu, select **Security > Security Groups**.
 - b) Select the group that provides authorization.
For example, select **FINANCE**.
 - c) Click the **GL Components** tab.
 - d) For each GL Component that is listed, select the **Authorized** check box.
As a shortcut, you can select **Authorize Group to Change All GL Component Types**.
 - e) Click **Save Group**.
10. Update the company-related accounts.
- a) From the **Go To** menu, select **Financials > Chart of Accounts**.
 - b) Select your organization and from the **More Actions** menu, select **Company-Related Accounts**.
 - c) Click **New Row** and specify the company type **Courier**.
 - d) Enter an account number in the **RBNI Account**, **AP Suspense Account**, and **AP Control Account** fields.
You can specify the same account number in each field.
 - e) Click **OK**.
 - f) From the **More Actions** menu, select **Update Database** and click **OK**.
11. Create a default insert site.
- a) From the **Go To** menu, select **Security > Users**.
 - b) Find **maxadmin** and select the record.
 - c) In the **Default Insert Site** field, specify the site name that you created in step 5.
 - d) In the **Storeroom Site for Self-Service Requisitions** field, specify the same site name.
 - e) Click **Save User**.
12. Define work types. Work types denote the importance of the work task.
- a) From the **Go To** menu, select **Administration > Organizations**.
 - b) Find your organization and select the record.
 - c) From the **More Actions** menu, select **Work Order Options > Work Type**.
 - d) Click **New Row**.
 - e) In the **Work Order Class** field, select **WORKORDER**.
 - f) Specify a **Work Type**.
For example, enter **MAJOR**.
 - g) Set the **Start Status** to **In Progress**.
 - h) Set the **Complete Status** to **Completed**.
 - i) Click **New Row** and repeat steps f-i to create another work order class with a different work type.
For example, enter **MINOR**.

- j) Click **New Row** and repeat steps f-i to create a CHANGE work order class with a different work type.
For example, enter SIG to represent a significant change.
- k) Click **OK** and click **Save Organization**.

Related information

[Integrating data with external applications](#)

[System properties](#)

Synchronizing users and groups

When you select application server security, the scheduled synchronization of users and groups that occurs between LDAP repositories and Maximo Asset Management is governed by the federated repositories.

Before you begin

View the cron task configuration information in the Administering section of the Knowledge Center.

About this task

LDAP repositories managed by IBM WebSphere Application Server Network Deployment through Virtual Member Manager are synchronized through the VMMSYNC cron task.

To configure the synchronization schedule between LDAP repositories and Maximo Asset Management, complete the following steps:

Procedure

1. Open a web browser and point to `http://host name:port/maximo`.
2. Log in to Maximo Asset Management using the maxadmin user ID.
3. From the Maximo Asset Management interface, navigate to **Go To > System Configuration > Platform Configuration > Cron Task Setup**.
4. Search for the appropriate cron task in the **Cron Task** field and configure it.
5. Set the task to **active**.

What to do next

By default, the cron task performs its task every 5 minutes. Change the **Schedule** field of the cron task if you want to change the interval.

Adding sample data to the Maximo Asset Management database post configuration

Using the **maxinst** command, you can add sample data to the Maximo Asset Management database after the initial installation and configuration of Maximo Asset Management.

Before you begin

The **maxinst** command does not provide default values for table space parameters. You must specify the data table space name and the index table space name that you specified during the installation. The **maxinst** command replaces the current database schema and data. Do not add sample data to the Maximo Asset Management database if you want to preserve records that are currently stored in the database. Running the **maxinst** command removes all post deployment configuration information and data in the Maximo Asset Management database.

About this task

If you created a database either automatically or manually during the installation, you can use the **maxinst** command to create a sample data in that database instance. You can add sample data to the database by selecting the option on the initial configuration panels or using the following steps:

Procedure

1. Back up the existing database.
2. Unzip the `install_home\IBM\SMP\Maximo\tools\maximo\en\maxdemo.dbtype.zip` file.
Replace the `dbtype` variable with the extension that is appropriate for the type of database you used for the Maximo Asset Management deployment. Available options are `ora`, `sqz`, or `db2`.
3. Change the `install_home\IBM\SMP\Maximo\tools\maximo\en\maxdemo.dbtype` file to the name of the Maximo Asset Management database you configured during the initial deployment.
For example, rename the file to `max76db.db2`
4. Open a command prompt and change to the `install_home\IBM\SMP\Maximo\tools\maximo` directory.
5. Populate the database with sample data. You must specify the table space name parameters when you run the **maxinst** command.

```
maxinst -stablespace -ttablespace -imax76db
```

For example, for Oracle Database or for Db2®, enter `maxinst -sMAXDATA -tMAXDATA -imax76db`.
For SQL Server, enter `maxinst -tPRIMARY -sPRIMARY -imax76db`.

The system reads the `maximo.properties` file for database connectivity information. The `maximo.properties` file is in the `install_home\IBM\SMP\Maximo\Applications\Maximo\Properties` directory.

6. Populate the database by running commands with specific parameter values. Use the following **maxinst** database parameters:

Parameter	Description
-a	Database alias. If not specified, the <code>mxedb.url</code> property is used.
-d	Log file directory. If you are using the <code>-l</code> parameter, the log file is sent to the specified directory. Otherwise, the log file is sent to the log directory, for example <code>install_home\IBM\SMP\Maximo\tools\maximo\logs</code> .
-e	Runs the SQL. This parameter is required and already present in the <code>maxinst.bat</code> file.
-f	File name for the properties file. If not specified, <code>maximo.properties</code> is used.
-i	File name of the input file (without path or extension).
-k	Directory of the properties file.
-l	Creates a detailed log file. This parameter is already present in the <code>maxinst.bat</code> file.
-m2	Optional value for SQL Server: Multiplies the column width for strings by 2. This value is required for double-byte character sets.
-o	If you are using the <code>-l</code> parameter, the <code>-o</code> parameter specifies the file name for the log file.
-p	Password for the database connection. If not specified, the <code>mxedb.password</code> property or <code>MAXIMO</code> is used. If <code>MAXIMO</code> is used, it must be entered in uppercase letters.
-s	Required value: Table space for index storage.
-t	Required value: Table space for table storage.
-u	User name for database connection. If not specified, the <code>mxedb.user</code> property or <code>MAXIMO</code> is used. If <code>MAXIMO</code> is used, it must be entered in uppercase letters.

Parameter	Description
-x	Required value for UNIX: Fixes the doclink file separators in UNIX environments. Note: If a UNIX environment is deployed without using this parameter, the attached documents do not function properly.
-y	Multiplies the column width for strings by 2. This value is required for double-byte character sets.

7. From the *install_home\IBM\SMP\ETC* directory, open the *install.properties* file and specify the installation-related properties.

You can add these properties to the database from the System Properties application.

Results

The Maximo Asset Management database is populated with sample data.

If an error occurs when you run the **maxinst** command, check the log file in the *install_home\IBM\SMP\Maximo\tools\maximo\logs* directory. If the **maxinst** command fails, you must correct the error and rerun the **maxinst** command to complete the Maximo database schema deployment before you can start the application.

Enabling best-practice links

Many start centers include a section called **My Policy Documents**, which contains a set of links to best-practice documents. These links open documents stored on the Tivoli® Unified Process Composer server. While this solution is optional, you must install or reuse and configure an existing Tivoli Unified Process Composer server to take advantage of this feature. If you do not implement this feature, you are not able to access best-practice documents hosted by a Tivoli Unified Process Composer server from a start center.

Before you begin

To enable these links you must complete these steps:

Procedure

1. Install Tivoli Unified Process Composer in your network. The Tivoli Unified Process Composer product image, including the installation instructions, is included on the Maximo Asset Management media.
2. Use the **updateStringInURLName** command to specify the host name where you installed Tivoli Unified Process Composer. This command is described in the topic "Specifying the Tivoli Unified Process Composer URL" in the Maximo Asset Management information center.

Chapter 7. Maintaining EAR files

This section contains information about maintaining Maximo Asset Management EAR files.

The following instructions are used to manually build or uninstall the Maximo Asset Management `maximo.ear` file on WebSphere Application Server Network Deployment.

Although the Maximo Asset Management installation program deploys these EAR files when you install, there might be a few instances where it would be desirable to rebuild these EAR files manually:

- If you modify any database connection parameters in the `maximo.properties` file after the initial installation, you must rebuild of the `maximo.ear` file, and then redeploy it in WebSphere Application Server Network Deployment. In this scenario, you would likely only rebuild and redeploy the `maximo.ear` file.
- Maximo Asset Management must be installed into a WebSphere Application Server Network Deployment application server. However, Maximo Asset Management can be run within the framework of a WebSphere Application Server Network Deployment cluster. If you want to deploy Maximo Asset Management in a cluster, you can either redeploy the Maximo Asset Management EAR files into a cluster, or, create a cluster from the application server used during the install. If you have already installed Maximo Asset Management into an application server but would like to redeploy into a cluster, then you must either uninstall the MAXIMO application (“[Manually uninstalling applications from WebSphere Application Server Network Deployment](#)” on page 18), or provide a new name for the application when installing the MAXIMO application into a cluster (“[Manually installing applications in WebSphere Application Server Network Deployment](#)” on page 18).
- If you have installed Maximo Asset Management into a development environment, you might at some point like to migrate the deployment into a test or production environment. In this scenario, you must deploy both the `maximo` and `maximo help` applications into the new environment.

The steps outlined in “[Manually installing applications in WebSphere Application Server Network Deployment](#)” on page 18 should be performed for both applications.

Building EAR files

You can manually build Maximo Asset Management EAR files if, for example, you modify a database connection parameter in the `maximo.properties` file after the initial installation.

About this task

To manually build Maximo Asset Management EAR files, use the `buildmaximoear` command.

Windows

```
install_home\maximo\deployment\buildmaximoear.cmd
```

Linux® and UNIX

```
install_home/maximo/deployment/buildmaximoear.sh
```

Manually deploying Work Center WAR files

The Work Center WAR file, `maximo-x.war`, is automatically built and deployed during the installation and configuration process. However, you can manually build the Work Center WAR file by using the **`buildmaximo-xwar`** command.

Procedure

1. Log on to the administrative server, open a command line, and change to the `maximo-root\deployment` directory.

- Run the **buildmaximo-xwar** command.
The `maximo-x.war` file is created in `maximo-root\deployment\default` directory.
- Deploy the `maximo-x.war` file on the Maximo Asset Management application server.

Configuration	Action
Simple	Deploy the <code>maximo-x.war</code> file on the Maximo Asset Management application server or UI cluster.
Proxy server that communicates with Maximo Asset Management by using a single hostname and port	Deploy the <code>maximo-x.war</code> file on a separate application server.
Cross origin requests to the Maximo Asset Management server	Configure the CORS settings by using servlet filters that set access control headers on the application server that hosts the <code>maximo-x</code> application.

Results

After you build and deploy the `maximo-x.war` file, use the default context path `/maximo-x` to access the Work Centers.

Manually uninstalling applications from WebSphere Application Server Network Deployment

This section contains information about uninstalling Maximo Asset Management applications from WebSphere Application Server Network Deployment.

Procedure

- Log in to the WebSphere Application Server Network Deployment administrative console, select **Servers > Server Types > WebSphere application servers**, select `MXServer` and click **Stop**.
- Click the Applications link.
- Select the check box next to the application you are uninstalling.
By default, the Maximo Asset Management applications are named `maximo` and `maximoiehs`.
- Click **Uninstall**.

Manually installing applications in WebSphere Application Server Network Deployment

Maximo Asset Management applications can be added to WebSphere Application Server Network Deployment.

Procedure

- Log on to the WebSphere Application Server Network Deployment administrative console.
- Browse to **Applications > New Application > New Enterprise Application**.
- From the Specify the EAR, WAR, JAR, or SAR module to upload and install page, select **Local file system**.
- Browse to the location on your system of the `maximo.ear` file and click **Next**.
- Select **Detailed** and then click **Next**.
- From the Application Security Warnings panel, click **Continue**.
- Click **Step 2: Map modules to servers**.
- Highlight all entries listed in the Clusters and servers field, check all check boxes for Modules listed in the table, and click **Apply**.
- Click **Step 11: Map virtual hosts for Web modules**.

10. Check all check boxes for web modules listed in the table.
11. Expand **Apply Multiple Mappings**.
12. Select a virtual host, for example, `maximo_host`, from the Virtual Host menu, and click **Apply**.
13. Click **Step 15: Map security roles to users or groups**.
14. Select the check box for `maximouser` in the Role table, and then select **Everyone** from the Map Special Subjects menu.
15. Click **Step 17: Summary**, review the summary information, and click **Finish**.

Chapter 8. Command line configuration

Maximo Asset Management includes a configuration program, **reconfigurePae**, located in the *Install_Home \ConfigTool\scripts* directory, that is accessed from the command line. This command line configuration program performs the same duties as the Maximo Asset Management configuration program user interface. Command line configuration is used to quickly update configuration values when you choose to not use the configuration tool user interface.

When you install a fix pack or upgrade Maximo Asset Management, the configuration program uses values that are recorded from the previous deployment. These values are stored on the administrative system in the *install.properties* and *maximo.properties* files that are located under the *install_home \etc* and *install_home\maximo\applications\maximo\properties* directories.

If you make any environmental changes to any of the systems that are used for the original deployment, they must be recorded in the *install.properties* and *maximo.properties* files. Typically, these types of changes would include changing a host name or updating a password, for example.

You can update these values manually, or you can use the configuration command line interface tool to update configuration values for your existing deployment when they change. Properties that are specified for the configuration command line interface tool, either as parameters, or in an input properties file, replace existing properties in the *maximo.properties* and *install.properties* files. Properties are encrypted when they are saved.

You can complete the following tasks with the configuration command line interface tool:

- Change the security model for the original deployment. For example, you can migrate from Maximo Asset Management-based security to WebSphere Application Server security. This option is not available for Oracle WebLogic Server.
- Confirm that updated property values conform to required input domains. Validate credentials, host names, and available ports.
- Update database or application server configuration settings without the server available. These values cannot be validated in this scenario.
- Clone an existing environment and use the configuration command line interface tool to reconfigure the administrative server to use the new servers. The cloned environment must use the same directory structure as the original.
- Create an environment (non-cloned) and use the configuration command line interface tool to create all required artifacts database, table spaces, and application server objects, for example.
- Remove configuration from a previously configured database.

Configuration parameter validation

Use the Maximo Asset Management command line configuration program to validate input in the same way the Maximo Asset Management configuration program validates input in the user interface.

You can validate the following items with the Maximo Asset Management command line configuration program.

- Host names or IP addresses are correctly formatted and are reachable.
- User IDs and passwords meet length and character set criteria.
- Specified credentials are used to authenticate to WebSphere Application Server.
- Specified port values are listening on the corresponding host. Port values contain numbers only and are within a valid range.
- Middleware installation directories that are supplied exist.
- Preconfigured middleware can authenticate to the product database.
- Required objects such as JMS queues, SIB destinations, databases, and table spaces exist.

- Required WebSphere Application Server virtual member manager users exist.
- Sufficient authority is available to create WebSphere Application Server virtual member manager users.
- Remote login credentials are valid.
- There is enough space in the supplied directories to create database objects.

Input properties files

You can use property files as input for the Maximo Asset Management command-line configuration program.

The following sample input property files are provided with the Maximo Asset Management command-line configuration program.

- `DB2_Sample_input.properties`
- `Oracle_WebLogic_Sample_Input.properties`
- `WebLogic_DB2_Sample_input.properties`
- `DB2_Websphere_MT_Sample_input.properties`
- `Oracle_Websphere_Sample_input.properties`
- `Websphere_App_Server_Sample_input.properties`
- `DB2_Websphere_Sample_Input.properties`
- `SQLServer_Sample_input.properties`
- `SQLServer_Weblogic_Sample_input.properties`
- `Oracle_Sample_input.properties`
- `SQLServer_Websphere_Sample_Input.properties`

These sample files are in the *InstallLocation*\ConfigTool\samples directory. Each of these sample files contains properties that are associated with a specific database or application server type. They contain comments and properties you can update with values from your environment. You can then use the updated file as input for the configuration command line interface tool with the **-input** parameter.

All input properties, introduced from either the command line or in the input properties file, are validated before they are used. After validation is successful, the `maximo.properties` and `install.properties` files are updated with the values you provided.

Command line interface parameters

The following parameters can be used with the configuration command line program to configure your environment.

Table 2. Maximo Asset Management command line configuration program parameters

Parameter name	Description
-action	<p>Configuration action.</p> <ul style="list-style-type: none"> • addLanguages – Adds additional languages to a current deployment. • configureProducts - Configures more products after an initial installation and configuration. This action should be used to configure multiple products after they are installed and should also be used in an upgrade scenario. • configureProductsDBLite - Configures more products after an initial installation and configuration using the DB Lite feature. This action can configure multiple products after they are installed and can also be used in an upgrade scenario. • deployConfiguration - Configures initial installation. This action can configure multiple products after they are installed. This action is used for new installations only. • deployDatabaseConfiguration - Validates command input and defines configuration property values for a new database for the current product. • deployJ2eeConfiguration - Validates command input and defines configuration property values for a new application server for the current product. • deployMiddlewareConfiguration – Configures initial middleware installation. • disableAppSecurity - Disables application security for the application. Security is handled entirely through Maximo Asset Management. This action sets the mxe.useAppServerSecurity property to a value of 0. This value is written to the database when updated. • disableDB2TextSearch - Disable the full text search feature on a Db2 database. • enableAppSecurity – Enables application security for the application. This action sets the mxe.useAppServerSecurity property to a value of 1, and updates the mxe.ldapUserManagement flag. This value is written to the database when the database is updated. • enableDB2TextSearch – Enables the full text search feature on a Db2 database. • reapplyAppServerTuning – Reapplies tuning configuration to an application server. • removeDatabaseConfiguration – Removes product configuration information from a previously configured database. • removeJ2EEConfiguration – Removes application server configuration that was added by the configuration tool. • updateApplication – Updates the Maximo Asset Management application by updating the database and redeploying application EAR files. This action should not be used in an upgrade scenario or when a new product is installed in an existing product.

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
	<ul style="list-style-type: none"> • updateApplicationDBLite – Update the database using the DB Lite feature and then rebuild and redeploy the EARs. • updateDatabaseConfiguration – Updates existing database configuration values. • updateJ2eeConfiguration - Updates existing application server configuration values. • upgrade75Configuration - Performs database properties validation for your Maximo Asset Management 7.5 deployment to ensure a valid connection to the database, and also configures WebSphere Application Server for Maximo Asset Management 7.6. When you upgrade from Maximo Asset Management 7.5, you must configure a new instance of WebSphere Application Server. • validateAndUpdateDatabaseConfiguration – Validates reconfiguration command line interface tool input and then updates existing database configuration property values. • validateAndUpdateJ2eeConfiguration - Validates reconfiguration command line interface tool input and then updates existing database configuration property values. • validateDatabaseConfiguration - Validates database configuration values specified as input for the reconfiguration command line interface tool. • validateJ2eeConfiguration - Validates application server configuration values that are specified as input for the reconfiguration command line interface tool. • removeConfiguration - Removes database and application configuration performed by the configuration tool. <p>The actions that modify the security setting for Maximo Asset Management also updates Maximo Asset Managementweb.xml files.</p>
-additionalLangs	<p>Adds one or more additional languages to the deployment.</p> <p>This parameter requires one or more locale abbreviations for input. For example, ES for Spanish.</p>
-allowUpdateDB	<p>The database update task is run to add the languages that are specified to the database.</p>
-applicationServerName	<p>Name of the application server.</p>
-applicationServerNode	<p>Name of the application server node.</p>
-automatedbconfig	<p>Automates the configuration of the database. Use this parameter to create the database instance, database users, the database, table spaces, and creates database schema.</p>
-automatej2eeconfig	<p>Automates the configuration of WebSphere Application Server. Using this parameter creates, for example, JMS queues and profiles.</p>

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
-buildAndDeployEAR	Rebuilds and deploys application EAR files. Application EAR files must be rebuilt and redeployed for configuration changes to take effect in the application.
-buildears	Rebuilds application EAR files.
-bypassJ2eeValidation	<p>Skips the validation and configuration of the application server.</p> <p>If you use this parameter, then you must manually configure the application server before you can deploy Maximo Asset Management.</p> <p>Validation of a manually configured application server requires logging in to the application server with application server administrative credentials. If you do not want to provide those credentials to the configuration program, you can skip the validation task.</p> <p>If you use the bypassJ2eeValidation parameter, you cannot also use the following parameters:</p> <ul style="list-style-type: none"> • applicationServerName • applicationServerNode • automatej2eeconfig • buildAndDeployEAR • createResourcesIfMissing • deploymaximoear • enableappsecurity • enableEnhancedNavigation • enableSkin • inputfile • j2eeserverport • usermanagement • wasuser • waspwd
-bypassUpdateDB	<p>Specifies that the database update task is not run and the installation program does not update the database.</p> <p>Languages files for the locales that are specified are built into the application EAR file.</p> <p>Use this parameter if you have run the database update task and included the language files in the database through another method.</p>
-createResourcesIfMissing	Creates any resources that are not found during configuration.
-db2_english_only	<p>Specifies that alphanumeric data is stored as a <code>varchar</code> data type. Using this parameter limits you to using English in the user interface.</p> <p>If you do not use this parameter, alphanumeric data is stored as a <code>varchargraphic</code> data type.</p>

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
-db2textsearchport	Port that is used by the Db2 server for full text search.
-dbname	Name of the database.
-dbpwd	Password for the user ID that accesses the database.
-dbrxapwd	Password for the user ID used to access a remote middleware server.
-dbrxauser	User ID used to access a remote middleware server.
-dbserverhost	Host name of the database server.
-dbserverport	Port name of the database server.
-dbuser	User ID that accesses the database.
-dbvendor	Database software name. <ul style="list-style-type: none"> • DB2 • Oracle • SQLServer
-deleteInstanceUsersAndGroups	Removes the instance owner and database user IDs and associated groups from the Db2 server. This parameter cannot be used with Microsoft SQL Server. The deleteInstanceUsersAndGroups parameter requires the dbrxauser and dbrxapwd and the dbuser and dbpwd parameters.
-deleteWASprofiles	Deletes the WebSphere Application Server profiles that are created for the application.
-deployDemoData	Includes sample data.
-deploymaximoear	Deploys Maximo application EAR file.
-enableappsecurity	Enables application security for the application. This parameter works the same as the enableAppSecurity action. This action sets the mxo.useAppServerSecurity property to a value of 1, and updates the mxo.ldapUserManagement flag. This value is written to the database when updated.
-enableEnhancedNavigation	Enables enhanced navigational elements in the user interface, including enhancements for switching between applications and viewing record lists. The enableEnhancedNavigation parameter must be used with the enableSkin parameter. If you enable enhanced navigation, and then you enable a different user interface skin with the -enableSkin parameter, you must re-enable enhanced navigation.

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
-enableMultiTenancy	<p>Enables the multitenancy feature.</p> <p>Multitenancy refers to software or services that can be configured to serve multiple groups of clients that are referred to as tenants. Each group of tenants is served by an instance of the software that uses data that only they can access.</p>
-enableRestrictivePasswordPolicy	<p>Enables a more restrictive password policy for Maximo Asset Management users.</p> <ul style="list-style-type: none"> • Login attempts <p>Standard policy allows 10 login attempts before the account is locked.</p> <p>Restrictive policy allows 3 login attempts before the account is locked.</p> • Number of times that the forgotten password link can be used. <p>Standard policy allows 5 uses of the forgotten password link before locking the account.</p> <p>Restrictive policy allows three uses of the forgotten password link before locking the account.</p> • Password duration (days) <p>Standard policy passwords never expire.</p> <p>Restrictive policy passwords expire after 90 days.</p> • Password expiration warning (days) <p>Standard policy does not warn the user that the account password is about to expire.</p> <p>Restrictive policy warns the user that the account password is about to expire seven days before it expires.</p> • Password threshold <p>Standard policy does not enforce a password threshold.</p> <p>Restrictive policy enforces a password threshold of 270 characters.</p>

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
	<ul style="list-style-type: none"> <li data-bbox="703 247 1464 373"> <p>• Password minimum character length</p> <p>Standard policy passwords must be at least six characters.</p> <p>Restrictive policy passwords must be at least eight characters.</p> <li data-bbox="703 384 1464 573"> <p>• Limit of allowed consecutive characters</p> <p>Standard policy has no restrictions to the amount of times a character can be repeated consecutively in a password.</p> <p>Restrictive policy allows a character to be repeated two times consecutively in a password value.</p> <li data-bbox="703 583 1464 772"> <p>• Password can be same value as user name</p> <p>Standard policy allows you to use the same value for the user name and password.</p> <p>Restrictive policy does not allow you to use the same value for the user name and password.</p> <li data-bbox="703 783 1464 972"> <p>• Number of numeric characters required in the password</p> <p>Standard policy does not require numeric characters in a password value.</p> <p>Restrictive policy requires one or more numeric characters in a password value.</p> <li data-bbox="703 982 1464 1171"> <p>• Number of alphabetic characters required in the password</p> <p>Standard policy does not require alphabetic characters in a password value.</p> <p>Restrictive policy requires one or more alphabetic characters in a password value.</p>

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
<p>-enableSkin</p>	<p>Change the appearance of the user interface.</p> <ul style="list-style-type: none"> • Classic <p>The following properties are set to 0:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs – mxe.webclient.verticalLabels <p>Sets the following property to null:</p> <ul style="list-style-type: none"> – mxe.webclient.skin • IoT18 <p>When enhanced navigation is enabled, the following properties are set to a value of 1:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs <p>The following property is set to a value of 2:</p> <ul style="list-style-type: none"> – mxe.webclient.verticalLabels <p>When enhanced navigation is not chosen, the following properties are set to a value of 0:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs – mxe.webclient.verticalLabels <p>Sets the following property:</p> <ul style="list-style-type: none"> – mxe.webclient.skin=IoT18 • tivoli09 <p>When enhanced navigation is enabled, the following properties are set to a value of 1:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs – mxe.webclient.verticalLabels <p>When enhanced navigation is not enabled, the following properties are set to a value of 0:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs – mxe.webclient.verticalLabels <p>Sets the following property:</p> <ul style="list-style-type: none"> – mxe.webclient.skin=tivoli09

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
	<ul style="list-style-type: none"> • tivoli13 <p>When enhanced navigation is enabled, the following properties are set to a value of 1:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs – mxe.webclient.verticalLabels <p>When enhanced navigation is not enabled, the following properties are set to a value of 0:</p> <ul style="list-style-type: none"> – mxe.webclient.homeButtonHeaders – mxe.webclient.systemNavBar – mxe.webclient.tabBreadCrumbs – mxe.webclient.verticalLabels <p>Sets the following property:</p> <ul style="list-style-type: none"> – mxe.webclient.skin=tivoli13
-force	<p>Indicates that an input property file updates the properties file and skips the validation of parameters input to the command line tool through an input properties file.</p> <p>When you use this parameter, you are not prompted for confirmation of the property update task.</p>
-inputfile	<p>Fully qualified path to the properties file that is used to set properties in an environment.</p> <p>Values that are specified as command line parameters for the configuration command line interface tool supersede values from the input file.</p>
-j2eeserverhost	<p>Host name of the domain manager of the application server.</p> <p>If deployment manager host name or server port is specified the thinwsadmin scripts are updated accordingly.</p>
-j2eeserverport	<p>Port name of the application server domain manager server.</p> <p>If deployment manager host name or server port is specified the thinwsadmin scripts are updated accordingly.</p>
-j2eevendor	<p>Application server software.</p> <ul style="list-style-type: none"> • WebSphere • WebLogic
-removeCreatedArtifacts	<p>Removes any artifacts that are created by the configuration tool. Examples include directories that are created by the configuration program, JMS objects, and the application server.</p>

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
-removeInstance	Removes the database instance for Db2 and Oracle. This parameter cannot be used with Microsoft SQL Server. The removeInstance parameter requires the dbrxouser and dbrxapwd , and the dbuser and dbpwd parameters.
-setjdbcurl	Sets the JDBC connection URL string.
-stopApplications	The Maximo Asset Management application server must be stopped when you add languages. If you do not use the stopApplications parameter with the addLanguages action, you must stop the application server manually before you use the addLanguages action. This parameter is used for WebSphere Application Server only.
-unsetjdbcurl	Removes the currently used JDBC connection URL string.
-updatedb	Runs the updatedb configuration task.
-usermanagement	Changes the security configuration of the existing Maximo deployment. <ul style="list-style-type: none"> • j2ee Change the security setting in Maximo so that both Maximo users and Maximo groups are managed through application server security mechanisms. • mixed Manage users through application server security mechanisms, and manage groups using Maximo.
-validateForNewDeploy	Validate middleware configuration. When this parameter is specified, validation is performed as if the tool was configuring the middleware. Additional configuration details are checked, including disk space availability and remote access credentials. If this parameter is not specified, validation is performed in the context of a component that is already established in the environment.
-validateUsers	Used with the enableAppSecurity action to validate users exist in the LDAP repository. If the users do not exist, the existing security model is not modified.
-waspwd	WebSphere Application Server administrator user ID password.
-wasrxapwd	Password for user ID to access remote WebSphere Application Server server host.
-wasrxouser	Operating system user ID used to access remote WebSphere Application Server host. Typically this user ID is defined either as Administrator or root.

Table 2. Maximo Asset Management command line configuration program parameters (continued)

Parameter name	Description
-wasuser	WebSphere Application Server administrator user ID. This is the user ID that is used to log in to the WebSphere Application Server administrative client application. Typically, this user ID is defined as wasadmin.

Command line configuration program actions

Use the command line configuration program to configure new or existing deployments of Maximo Asset Management.

action updateDatabaseConfiguration

The **updateDatabaseConfiguration** action is used to update existing database configuration values.

```
-action updateDatabaseConfiguration [-force] [-buildAndDeployEAR]
[-dbserverhost hostname] [-dbserverport port number]
[-dbname database name] [-dbuser userid]
[-dbpwd password] [-dbixauser userid]
[-dbixapwd password] [-setjdbcurl jdbcurl]
[-unsetjdbcurl] [-inputfile path to input properties file]
```

action validateDatabaseConfiguration

Use the **validateDatabaseConfiguration** action to validate current database configuration properties. This action can be used to verify database properties before they are updated, or following an update action.

```
-action validateDatabaseConfiguration [-validateForNewDeploy]
[-dbserverhost hostname]
[-dbserverport port number]
[-dbname database name]
[-dbuser userid] [-dbpwd password]
[-dbixauser userid]
[-dbixapwd password] [-setjdbcurl jdbcurl]
[-unsetjdbcurl] [-enableMultiTenancy]
[-inputfile path to input properties file]
```

The following properties are required if the **-validateForNewDeploy** parameter is used.

- **Database.DB2.InstanceAdminGroup**
- **Database.DB2.InstanceAdminUserName** (Windows only)
- **Database.DB2.InstanceAdminPassword** (Windows only)
- **Database.DB2.FencedUser**
- **Database.DB2.FencedUserPassword**
- **Database.DB2.FencedGroupName**
- **Database.DB2.InstanceUserPassword**

These properties must be added manually to the `install.properties` file.

action validateAndUpdateDatabaseConfiguration

The **validateAndUpdateDatabaseConfiguration** action is used to both validate reconfiguration command-line interface tool input and then update existing database configuration property values.

```
-action validateAndUpdateDatabaseConfiguration [-validateForNewDeploy] [-force]
[-buildAndDeployEAR] [-dbserverhost hostname]
[-dbserverport port number] [-dbname database name]
[-dbuser userid] [-dbpwd password]
[-dbixauser userid] [-dbixapwd password]
```

```
[-setjdbcurl jdbcurl] [-unsetjdbcurl] [-enableMultiTenancy]
[-inputfile path to input properties file]
```

The following properties are required if the **-validateForNewDeploy** parameter is used.

- **Database.DB2.InstanceAdminGroup**
- **Database.DB2.InstanceAdminUserName** (Windows only)
- **Database.DB2.InstanceAdminPassword** (Windows only)
- **Database.DB2.FencedUser**
- **Database.DB2.FencedUserPassword**
- **Database.DB2.FencedGroupName**
- **Database.DB2.InstanceUserPassword**

These properties that must be added manually to the `install.properties` file.

The **-enableMultiTenancy** parameter must be used with the **-validateForNewDeploy** parameter.

action deployDatabaseConfiguration

The **deployDatabaseConfiguration** action is used to validate command-line configuration program input and define configuration property values for a new database for the currently deployed product. This action can be run after a successful **deployConfiguration** action.

```
-action deployDatabaseConfiguration [-buildAndDeployEAR] [-createResourcesIfMissing]
[-dbserverhost hostname] [-dbserverport port number]
\[-dbname database name] [-dbuser userid]
[-dbpwd password] [-dbxouser userid] [-dbxapwd password]
[-setjdbcurl jdbcurl] [-unsetjdbcurl] [-enableMultiTenancy]
[-deployDemoData]
[-inputfile path to input properties file] [-enableRestrictivePasswordPolicy]
```

The following properties are required if the **-createResourcesIfMissing** parameter is used.

- **Database.DB2.InstanceAdminGroup**
- **Database.DB2.InstanceAdminUserName** (Windows only)
- **Database.DB2.InstanceAdminPassword** (Windows only)
- **Database.DB2.FencedUser**
- **Database.DB2.FencedUserPassword**
- **Database.DB2.FencedGroupName**
- **Database.DB2.InstanceUserPassword**

These properties that must be added manually to the `install.properties` file.

action removeDatabaseConfiguration

The **removeDatabaseConfiguration** action is used to remove configuration settings from a previously configured database.

```
-action removeDatabaseConfiguration [-removeInstance] [-dbxouser userid]
[-dbxapwd password] [-dbuser userid]
[-dbpwd password] [-deleteInstanceUsersAndGroups]
```

action enableDB2TextSearch

The **enableDB2TextSearch** action is used to enable the full text search feature for a Db2 database.

```
-action enableDB2TextSearch [-db2textsearchport port number]
```

action updateJ2eeConfiguration

The **updateJ2eeConfiguration** action is used to update existing application server configuration values.

```
-action updateJ2eeConfiguration [-force] [-buildAndDeployEAR]
[-j2eeserverhost hostname] [-j2eeserverport port number]
[-wasuser userid] [-waspwd password]
[-applicationServerNode nodeName] [-applicationServerName appServerName]
[-inputfile path to input properties file]
```

action validateJ2eeConfiguration

The **validateJ2eeConfiguration** action is used to validate application server configuration values specified as input for the reconfiguration command-line interface tool.

```
-action validateJ2eeConfiguration [-validateForNewDeploy] [-j2eeserverhost hostname]
[-j2eeserverport port number] [-wasuser userid]
[-waspwd password] [-applicationServerNode nodeName]
[-applicationServerName appServerName] [-inputfile path to input properties file]
```

The following properties are required if the **-validateForNewDeploy** parameter is used and the **WAS.SibPersistMessages** property is set to true.

- **Database.DB2.FencedUser**
- **Database.DB2.FencedUserPassword**
- **Database.DB2.InstanceUserPassword**

These properties that must be added manually to the `install.properties` file.

action validateAndUpdateJ2eeConfiguration

The **validateAndUpdateJ2eeConfiguration** action is used to both validate reconfiguration command-line interface tool input and then update existing application server configuration property values.

```
-action validateAndUpdateJ2eeConfiguration [-force] [-validateForNewDeploy]
[-buildAndDeployEAR] [-j2eeserverhost hostname]
[-j2eeserverport port number] [-wasuser userid]
[-waspwd password] [-applicationServerNode nodeName]
[-applicationServerName appServerName]
[-inputfile path to input properties file]
```

The following properties are required if the **-validateForNewDeploy** parameter is used and the **WAS.SibPersistMessages** property is set to true.

- **Database.DB2.FencedUser**
- **Database.DB2.FencedUserPassword**
- **Database.DB2.InstanceUserPassword**

These properties must be added manually to the `install.properties` file.

action deployJ2eeConfiguration

The **deployJ2eeConfiguration** action is used to validate command line configuration program input and define configuration property values for a new application server for the currently deployed product. This action can be run after a successful **deployConfiguration** action.

```
-action deployJ2eeConfiguration [-bypassJ2eeValidation] [-buildAndDeployEAR]
[-createResourcesIfMissing] [-j2eeserverhost hostname]
[-j2eeserverport port number] [-wasuser userid]
[-waspwd password] [-applicationServerNode nodeName]
[-applicationServerName appServerName]
[-inputfile path to input properties file]
[-enableSkin Classic|tivoli09|tivoli13] [-enableEnhancedNavigation]
```

The following properties are required if the **-validateForNewDeploy** parameter is used and the **WAS.SibPersistMessages** property is set to true.

- **Database.DB2.FencedUser**
- **Database.DB2.FencedUserPassword**
- **Database.DB2.InstanceUserPassword**

These properties that must be added manually to the `install.properties` file.

action removeJ2EEConfiguration

The **removeJ2EEConfiguration** action is used to remove configuration settings from a previously configured application server. Items removed include domain manager and application server profiles, and JMS objects.

```
-action removeJ2EEConfiguration [-wasuser userid] [waspwd password]  
[-wasrxouser userid] [-wasrxapwd password]  
[-removeCreatedArtifacts] [-deleteWasProfiles]
```

action deployConfiguration

Use the **deployConfiguration** action to configure the initial installation. This action is used for new Maximo Asset Management installation scenarios only. This action can configure multiple middleware products after they are installed. The **deployConfiguration** action requires the **inputfile** parameter and an input properties file populated with the configuration values you want to use.

```
-action deployConfiguration [-inputfile path to input properties file]  
[-dbvendor Oracle|DB2|SQLServer] [-j2eevendor WebSphere|WebLogic]  
[-bypassJ2eeValidation] [-automatedbconfig] [-automatej2eeconfig]  
[-usermanagement j2ee|mixed] [-buildears] [-deploymaximoear]  
[-enableappsecurity] [-deployDemoData] [-enableMultiTenancy] [-db2_english_only]  
[-enableSkin Classic|tivoli09|tivoli13] [-enableEnhancedNavigation]  
[-enableRestrictivePasswordPolicy]
```

action removeConfiguration

The **removeConfiguration** action is used to removes database and application configuration that is performed by the configuration tool. This action is equivalent to running the **removeJ2EEConfiguration** and **removeDatabaseConfiguration** actions together

```
-action removeConfiguration -dbuser userid  
-dbpwd password -wasuser userid  
-waspwd password [-wasrxouser userid]  
[-wasrxapwd password] [-dbrxouser userid]  
[-dbrxapwd password] [-removeInstance] [-deleteInstanceUsersAndGroups]  
[-removeCreatedArtifacts] [-deleteWasProfiles]  
[-inputfile path to input properties file]
```

action reapplyAppServerTuning

The **reapplyAppServerTuning** action is used to reapply best practices application server tuning settings.

```
-action reapplyAppServerTuning [-wasuser userid]  
[-waspwd password] [-applicationServerNode nodeName]  
[-applicationServerName appServerName]
```

You must restart the application server after the settings are applied.

action enableAppSecurity

The **enableAppSecurity** action is used to enable application security for the application. This action sets the **mxe.useAppServerSecurity** property to a value of 1, and updates the **mxe.ldapUserManagement** property according to the setting of the **-usermanagement** parameter.

This value is written to the database when updated. The maximouiweb, maxrestweb, meaweb, and mboweb web.xml files are updated during this action.

```
-action enableAppSecurity -usermanagement {j2ee,mixed}  
[-buildAndDeployEAR] [-validateUsers] [-force]
```

action disableAppSecurity

The **disableAppSecurity** action is used to disable application security for the application. This action sets the **mxo.useAppServerSecurity** property to a value of 0. This value is written to the database when updated. Using this action reverts the security implementation to Maximo security for users and groups. The maximouiweb, maxrestweb, meaweb, and mboweb web.xml files are updated during this action.

```
-action disableAppSecurity [-buildAndDeployEar] [-force]
```

action updateApplication

The **updateApplication** action is used to update the application by updating the database and redeploying application EAR files. This action should not be used in an upgrade scenario or when a new product is installed in an existing product.

```
-action updateApplication [-updatedb] [-deploymaximoear]  
[-enableSkin Classic|tivoli09|tivoli13] [-enableEnhancedNavigation]  
[-enableRestrictivePasswordPolicy]
```

action addLanguages

Use the **addLanguages** action to configure additional languages for a deployment.

```
-action addLanguages [-additionalLangs locale1,locale2,locale3]  
[-buildAndDeployEAR] [-allowUpdateDB | -bypassUpdateDB] [-stopApplications]  
[-inputfile path to input properties file]
```

The application server must be stopped when you add languages. If you do not use the **stopApplications** parameter with the **addLanguages** action, you must stop the application server manually before you use the **addLanguages** action.

The following locale values are supported.

Language	Locale
Arabic	ar
Brazilian Portuguese	pt_BR
Croatian	hr
Czech	cs
Danish	da
Dutch	nl
Finnish	fi
French	fr
German	de
Hebrew	he
Hungarian	hu

Table 3. Locales supported by the configuration program (continued)

Language	Locale
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Polish	pl
Russian	ru
Simplified Chinese	zh_CN
Slovak	sk
Slovenian	sl
Spanish	es
Swedish	sv
Traditional Chinese	zh_TW
Turkish	tr

action configureProducts

The **configureProducts** action should be used to configure more products that are installed after an initial installation and configuration. This action verifies a full installation and configuration of an existing product was completed successfully before it continues. This action can configure multiple products after they are installed. This action should be used in an upgrade scenario.

```
-action configureProducts [-updatedb] [-buildears] [-deploymaximoear]
```

Note: The EAR file is both built and deployed when you specify either the `-buildears` or the `-deploymaximoear` attribute.

action deployMiddlewareConfiguration

The **deployMiddlewareConfiguration** action is used to configures middleware after it is installed.

```
-action deployMiddlewareConfiguration [-inputfile path to input properties file]
```

The following properties can be included in the input properties file used with the **deployMiddlewareConfiguration** action.

Table 4. Properties that can be used in the `deployMiddlewareConfiguration` action input file

Category	Property
HTTP Server properties	IHS.HTTPPort
	IHS.InstallLocation
	IHS.WebserverName
LDAP server properties	LDAP.AdminDN
	LDAP.AdminPassword
	LDAP.BaseEntry
	LDAP.GroupSuffix

Table 4. Properties that can be used in the `deployMiddlewareConfiguration` action input file (continued)

Category	Property
	LDAP.OrgContainerSuffix
	LDAP.ServerHostName
	LDAP.ServerPort
	LDAP.UserSuffix
	LDAP.Vendor
General middleware configuration properties	MW.Operation
WebSphere properties	PLG.InstallLocation
	WAS.AdminPassword
	WAS.AdminUserName
	WAS.ApplicationServerName
	WAS.CellName
	WAS.DeploymentManagerNodeName
\	WAS.DeploymentManagerProfileName
	WAS.InstallLocation
	WAS.LDAPAutomatedConfig
	WAS.ND.AutomateConfig
	WAS.NodeName
	WAS.ServerProfileName
	WAS.SOAPConnectorPort
	WCT.InstallLocation

action upgrade75Configuration

The **upgrade75Configuration** action performs database properties validation for your Maximo Asset Management 7.5 deployment to ensure a valid connection to the database, and also configures WebSphere Application Server for Maximo Asset Management 7.6. When you upgrade from Maximo Asset Management 7.5, you must configure a new instance of WebSphere Application Server.

```
-action upgrade75Configuration [-inputfile path to input properties file]
[-dbvendor <Oracle|DB2|SQLServer>] [-j2eevendor <WebSphere|WebLogic>]
[-automatej2eeconfig] [-usermanagement <j2ee|mixed>]
[-enableappsecurity] [-bypassJ2eeValidation]
```

action updateApplicationDBLite

The **updateApplicationDBLite** action is used to update the database using the DB Lite feature and then rebuild and redeploy the EARs. This action should not be used in an upgrade scenario or when a new product is installed in an existing product.

```
-action updateApplicationDBLite [-preprocessor] [-updatedb]
[-buildears] [-deploymaximoear] [-enableSkin tivoli09|tivoli13|classic]
[-enableEnhancedNavigation] [-disableEnhancedNavigation]
[-enableRestrictivePasswordPolicy]
```

The `-preprocessor` attribute can be used to prepare class files before they are built into the application EAR files. The non-lite version of the `updatedb` command can also perform this work, however, it requires

that you stop the application server. If you use the `-preprocessor` attribute, you are not required to stop the application server while class files are prepared, decreasing the amount of downtime for your environment.

The `-updatedb` attribute can be used to update the database independently. When you use this attribute, the application server must be stopped. The `updatedb` process does not include preparing class files, which shortens the amount of time that your application server is down.

The `-preprocessor` and `-updatedb` attributes are mutually exclusive. You cannot use them together. The `-buildears` attribute can only be used with the `-preprocessor` attribute. The `-deploymaximoeear` attribute can only be used with the `-updatedb` attribute. Therefore, you must run the action twice to complete the configuration.

To prepare class files and build the EAR file, enter the following:

```
-action configureProductsDBLite -preprocessor -buildears
```

To update the database and deploy the EAR file, enter the following:

```
-action configureProductsDBLite -updatedb -deploymaximoeear
```

action configureProductsDBLite

The **configureProductsDBLite** action should be used to configure products after an initial installation and configuration using the DB Lite feature. This action can configure multiple products after they are installed and should be used in an upgrade scenario.

```
-action configureProductsDBLite [-preprocessor] [-updatedb]  
[-buildears] [-deploymaximoeear]
```

The `-preprocessor` attribute can be used to prepare class files before they are built into the application EAR files. The non-lite version of the `updatedb` command can also perform this work, however, it requires that you stop the application server. If you use the `-preprocessor` attribute, you are not required to stop the application server while class files are prepared, decreasing the amount of downtime for your environment.

The `-updatedb` attribute can be used to update the database independently. When you use this attribute, the application server must be stopped. The `updatedb` process does not include preparing class files, which shortens the amount of time that your application server is down.

The `-preprocessor` and `-updatedb` attributes are mutually exclusive. You cannot use them together. The `-buildears` attribute can only be used with the `-preprocessor` attribute. The `-deploymaximoeear` attribute can only be used with the `-updatedb` attribute. Therefore, you must run the action twice to complete the configuration.

To prepare class files and build the EAR file, enter the following:

```
-action configureProductsDBLite -preprocessor -buildears
```

To update the database and deploy the EAR file, enter the following:

```
-action configureProductsDBLite -updatedb -deploymaximoeear
```

Configuration program properties

Maximo Asset Management properties can be modified with the command line configuration program by using an `input.properties` file. When you set a property value to a Windows path, you must use two consecutive backslashes, for example, `C:\\IBM\\WebSphere\\AppServer`. When you set property values to a Linux or UNIX path, use single forward slashes. For example, `/opt/IBM/db2/V10.5`.

Table 5. Maximo Asset Management properties

Category	Property	Definition
Shared database properties	mxe.db.schemaowner	Owner of the database schema. For example, maximo. This value is written to the database when updated.
	mxe.db.user	User ID that accesses the database.
	mxe.db.password	Password for the user ID that accesses the database.
	Database.RemoteAccessUserName	Database server system user ID that is used for configure the database remotely. This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.
	Database.RemoteAccessPassword	Password for user ID named in Database.RemoteAccessUserName . This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.
Db2 Properties	Database.DB2.ServerHostName	Fully qualified host name of the Db2 server. For example, mymachine.mydomain.com. This value is written to the database when updated.
	Database.DB2.ServerPort	Database server port. For example, 50005. This value is written to the database when updated.
	Database.DB2.InstanceName	Name of the database instance. For example, ctginst1. This value is written to the database when updated. This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.DB2.InstanceUserPassword	<p>Password for the database instance owner</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.InstanceAdminGroup	<p>Group for the instance administrator. For example, ctgiadm1.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.DatabaseName	<p>Name of the database. For example, maxdb76.</p> <p>This value is written to the database when updated.</p>
	Database.DB2.InstallLocation	<p>Install location of the database. For example, /opt/IBM/db2/V10.5.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.LogFileSize	<p>Set the size for transaction logs. For example, 8192.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.AppCtlHeapSize	<p>Application control heap size. For example, 1024.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.DB2.ApplHeapSize	<p>Application heap size. For example, 1024.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.LockListSize	<p>Size that is allocated to the lock list. For example, AUTOMATIC.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.LogSecond	<p>Number of secondary log files allowed. For example, 100.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.ServiceUser	<p>User ID used to autostart.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p> <p>This property is only required if Db2 is installed on a Windows system.</p>
	Database.DB2.ServicePassword	<p>Password for Database.DB2.ServiceUser.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p> <p>This property is only required if Db2 is installed on a Windows system.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.DB2.ExtentSize	<p>Number of pages per extent (group of pages).</p> <p>For example, 32.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.FencedUser	<p>Fenced user ID for Db2 on Linux or UNIX systems.</p> <p>For example, db2fenc1.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.FencedUserPassword	<p>Password for the fenced user ID for Db2 on Linux or UNIX systems.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.FencedGroupName	<p>Default group for database fenced user.</p> <p>For example, ctgfgrp1.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p> <p>This property is only required if Db2 is installed on a Linux or UNIX system.</p>
	Database.DB2.AuthType	<p>Method Db2 uses to authenticate users.</p> <p>For example, server.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.DB2.DataTablespaceName	<p>Db2 table space name for the product database.</p> <p>For example, maxdata.</p>
	Database.DB2.BufferPoolName	<p>Db2 buffer pool name.</p> <p>For example, MAXBUFFPOOL.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.BufferPoolSize	<p>Size of the buffer pool.</p> <p>For example, 4096.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.DataTablespaceLocation	<p>Location of Db2 database table space data files.</p> <p>For example, CTGDAT.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.DataTablespaceSize	<p>Table space size, which is measured in Mb.</p> <p>For example, 5000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.TempTablespaceName	<p>Temporary table space name.</p> <p>For example, maxtemp.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.DB2.TempTablespaceLocation	<p>Location of temporary table space. For example, CTGTMP.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.TempTablespaceSize	<p>Temporary table space size, which is measured in Mb. For example, 1000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.IndexTablespaceName	<p>Index table space name. For example, maxdata.</p>
	Database.DB2.IndexTablespaceLocation	<p>Location of index table space. For example, CTGDAT.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.IndexTablespaceSize	<p>Index table space size, which is measured in Mb. For example, 5000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.DB2.InstanceAdminUserName	<p>Administrative user or the database instance. For example, db2admin.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p> <p>For Unix and Linux platforms, this value must be the same as the instance owner.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.DB2.InstanceAdminPassword	<p>Password for the user ID specified for Database.DB2.InstanceAdminUserName.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p> <p>createResourcesIfMissing</p>
	Database.Windows.AdminGroup.name	Windows administration group name for all users that requires administrative privileges.
Oracle	Database.Oracle.SchemaPassword	Password for the schema owner.
	Database.Oracle.InstanceName	<p>Oracle instance name.</p> <p>This value is written to the database when updated.</p>
	Database.Oracle.SoftwareOwner	<p>Owner of the software installation.</p> <p>For example, oracle.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.SoftwareOwnerPassword	<p>Password for the user ID listed in Database.Oracle.SoftwareOwner.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.InstallLocation	<p>Oracle installation location. For example, /opt/app/oracle/product/10.2.0/db_1.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.DataTablesapceName	<p>Oracle table space name for the product database.</p> <p>For example, maxdata.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.Oracle.InstanceLocation	<p>Oracle instance location.</p> <p>For example, /opt/app/oracle/product/10.2.0/db_1.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.DataTablespaceSize	<p>Table space size, which is measured in Mb.</p> <p>For example, 5000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.TempTablespaceName	<p>Temporary table space name.</p> <p>For example, maxtemp.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.TempTablespaceSize	<p>Temporary table space size, which is measured in Mb.</p> <p>For example, 1000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.IndexTablespaceName	<p>Index table space name.</p> <p>For example, maxdata.</p>
	Database.Oracle.IndexTablespaceSize	<p>Index table space size, which is measured in Mb.</p> <p>For example, 1000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.Oracle.ServerHostName	<p>Fully qualified host name of the Oracle server.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.Oracle.ServerPort	Port number that is used by Oracle. For example, 1521. This value is written to the database when updated.
	Database.DBUserName	Oracle DBA user name. For example, sys. This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.
	Database.DBAPassword	Password for user ID listed for Database.DBUserName . This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.
Microsoft SQL Server	Database.SQL.DatabaseName	Name of the database. For example, maxdb76. This value is written to the database when updated.
	Database.SQL.DataFileName	A way to specify the name of the data file used for the database. For example, maxdb76_dat. This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.
	Database.SQL.DataFileSize	Initial size for data file for the database in kb. For example, 5000. This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.SQL.LogFileName	<p>A way to specify the name for the database transaction log file.</p> <p>For example, maxdb76_log.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.SQL.LogFileSize	<p>Microsoft SQL Server transaction log file size.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.SQL.DataFilegroupName	<p>Database logical name file group.</p> <p>For example, PRIMARY.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>
	Database.SQL.ServerHostName	<p>Host name of the database server.</p> <p>For example, myhost.mydomain.com.</p>
	Database.SQL.ServerPort	<p>Database server port.</p> <p>For example, 1433.</p> <p>This value is written to the database when updated.</p>
	Database.SQL.InstanceAdminUserName	<p>Administrative user for the Microsoft SQL Server instance. Used during installation for creating and modifying the database and database user.</p> <p>For example, sa.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatedbconfig parameters are specified.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	Database.SQL.InstanceAdminPassword	Administrative user password. This property is only required if the createResourcesIfMissing , validatefornewdeploy , or automatedbconfig parameters are specified.
WebSphere Application Server Network Deployment	WAS.InstallLocation	Installation location for WebSphere Application Server Network Deployment. For example, C:\IBM\WebSphere\AppServer. This property is required if the WAS.SibPersistMessages property is set to true.
	WAS.DeploymentManagerHostName	Host name of the WebSphere Application Server Network Deployment deployment manager. This value is written to the database when updated.
	WAS.DeploymentManagerProfileName	WebSphere Application Server Network Deployment profile name. For example, ctgDmgr01. This property is required if the WAS.SibPersistMessages property is set to true.
	WAS.DeploymentManagerProfileRoot	Location of the WebSphere Application Server Network Deployment profile. For example, C:\IBM\WebSphere\AppServer\profiles\ctgDmgr01 This property is only required if the WAS.SibPersistMessages property is set to true.
	WAS.NodeName	WebSphere Application Server Network Deployment node name. For example, ctgNode01
	WAS.ApplicationServerName	WebSphere Application Server Network Deployment application server name. For example, MXServer. This value is written to the database when updated.

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	WAS.CellName	WebSphere Application Server Network Deployment cell name. For example, ctgCell01.
	WAS.AdminUserName	WebSphere Application Server Network Deployment administrator name. For example, wasadmin.
	WAS.AdminPassword	WebSphere Application Server Network Deployment administrator password.
	WAS.RemoteAccessUserName	WebSphere Application Server Network Deployment deployment manager system user ID used for tasks such as copying Integrated Solutions Console web archive files and fetching the keystore. This property is required if the WAS.SibPersistMessages property is set to true.
	WAS.RemoteAccessPassword	WebSphere Application Server Network Deployment deployment manager system user password. This property is required if the WAS.SibPersistMessages property is set to true.
	WAS.SOAPConnectorPort	SOAP port for WebSphere Application Server Network Deployment deployment manager. For example, 8879.
	WAS.VirtualHost	Name of the WebSphere Application Server Network Deployment virtual host. For example, maximo_host.
	WAS.DeploymentTopologyInputFile	Name of the WebSphere Application Server Deployment Manager topology JSON file. This property is required if the WAS.UseClustersForDeploy is set to true.
	WAS.WebServerHostName	Host name where the HTTP server is located.
	WAS.SibName	Name of the service integration bus. For example, intjmsbus.

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	WAS.SibHiMsg	<p>Service integration bus high message count.</p> <p>For example, 500000.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatej2eeconfig parameters are specified.</p>
	WAS.WebServerName	<p>Name of the WebSphere Application Server Network Deployment web server. Used to manage HTTP server from within WebSphere Application Server Network Deployment.</p> <p>For example, webserver1.</p>
	WAS.SibPersistMessages	<p>Binary value that indicates if service integration bus messages are persisted in either the product database or a local derby database.</p> <p>A value of true indicates that the messages are persisted.</p> <p>This property is only required if the createResourcesIfMissing, validatefornewdeploy, or automatej2eeconfig parameters are specified.</p>
	WAS.SibDSName	<p>Service integration bus data source name that is created to access the service integration bus persistence store.</p> <p>For example, intjmsds.</p> <p>This property is only required if the WAS.SibPersistMessages property is set to true.</p>
	WAS.SibDBName	<p>Name of the service integration bus messages database.</p>
	WAS.SibDBInstance	<p>Instance name of the service integration bus database.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	WAS.SibDBInstancePassword	<p>IBM Db2 instance users password for the SIB IBM Db2 data store.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>
	WAS.SibDBServerName	<p>Server name of the system that is hosting the service integration bus message database.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>
	WAS.SibDBServerPort	<p>Database server port for the database that contains the service integration bus messages.</p> <p>For example, 50005.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>
	WAS.SibDBUserName	<p>User ID used to access the persistence data store database for service integration bus messages.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>
	WAS.SibDBUserPass	<p>Password for user ID named in WAS.SibDBUserName.</p> <p>This property is only required if the WAS.SibPersistMessages property is set to true.</p>
	WAS.SibDBInstallDir	<p>Where the service integration bus database is installed.</p> <p>For example, c:\\program files \\ibm\\sqllib.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	WAS.SibDbFencedUser	<p>Fenced user ID for the service integration bus database. This property is only used for databases that are hosted on Linux and UNIX systems.</p> <p>For example, db2fenc1.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p> <p>This property is required if the database is hosted on a Linux or UNIX system.</p>
	WAS.SibDbFencedPassword	<p>Password for the fenced user ID for the service integration bus database.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p> <p>This property is required if the database is hosted on a Linux or UNIX system.</p>
	WAS.SibDbInstanceAdminUser	<p>Instance owner for the service integration bus database.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p> <p>This property is required if the database is hosted on a Windows system.</p>
	WAS.SibDbInstanceAdminPassword	<p>Password for the instance owner of the service integration bus database.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p> <p>This property is required if the database is hosted on a Windows system.</p>
	WAS.SibDbRemoteAccessUser	<p>Database server system user that is used to configure the service integration bus remotely.</p> <p>This property is required if the WAS.SibPersistMessages property is set to true.</p>

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	WAS.SibDbRemoteAccessPassword	Password for user ID named in WAS.SibDbRemoteAccessUser . This property is required if the WAS.SibPersistMessages property is set to true.
	WAS.VmmGroupRDN	Relative Distinguished Name of the location of virtual member manager groups. For example, ou=groups,ou=SWG,o=IBM,c=US.
	WAS.VmmUserRDN	Relative Distinguished Name of the location of virtual member manager users. For example, ou=users,ou=SWG,o=IBM,c=US.
	WAS.UseClustersForDeploy	Deploy the application EAR to a cluster. For example, true.
Multi-tenancy	mt.sysprovider.tenantcode	Tenant code identifier for the multi-tenancy system provider. For example, MTM. This property is required to enable multi-tenancy.
	mt.sysprovider.desc	Description of the multi-tenancy manager.
	mt.configmanager.tenantcode	Tenant code identifier for the multi-tenancy configuration manager. For example, GA. This property is required to enable multi-tenancy.
	mt.configmanager.desc	Description of the multi-tenancy global administrator.
	mt.configmanager.dbuser	User ID of the multi-tenancy configuration manager used to connect to the database. For example, ADMIN. This property is required to enable multi-tenancy. This value can be the same value used for the mtdbuser user.

Table 5. Maximo Asset Management properties (continued)

Category	Property	Definition
	mt.configmanager.dbpassword	<p>Password for the user ID of the multi-tenancy configuration manager that is used to connect to the database.</p> <p>This property is required to enable multi-tenancy.</p> <p>This value can be the same value used for the mtdbpwd value.</p>
	mxe.mt.db.adminrole	<p>Admin role that has permission to access template data when tenants are initially provisioned.</p> <p>The default value is MXTMPLT. However, users can create a different role and specify that value for the property.</p>
	mxe.mt.demo.extratenants	<p>Number of sample tenants to create when you enable multi-tenancy.</p> <p>For example, 10.</p> <p>This property is set in the <code>maximo.properties</code> file during the maxinst operation.</p>

Chapter 9. Backup and restoration

Like all important business data, it is a good idea to establish a process and schedule for backing up Maximo Asset Management data.

Back up and restore middleware application data by using the methods that are described in the documentation for that product is important. Also, establish a process for backing up data that is contained on the Maximo Asset Management administrative server.

The default installation directory on the Maximo Asset Management administrative server is C:\ibm. This directory contains the critical files for your Maximo Asset Management deployment, which includes all class files and customizations that are made to your environment, the current Enterprise Archive (EAR) file, and the properties files and installation tools used for your environment. Plan to back up Maximo Asset Management administrative server data just after initial deployment and schedule periodic backups on an ongoing basis.

Backing up the administrative server

It is recommended that you back up all Maximo Asset Management middleware applications and Maximo Asset Management administrative server regularly.

About this task

The default installation directory on the administrative server is \ibm. This directory contains the critical files for your Maximo Asset Management deployment.

Specifically, the administrative server contains the following items:

- Class files and customizations that are performed in your environment.
- The current Enterprise Archive (EAR) file that was deployed to the application server.
- The properties files and installation tools that are used in your environment.

It is important to make a backup of the database at the same time that you back up the administrative server. During restoration, you restore the database backup at the same time you restore the administrative server backup it was paired with.

To back up critical Maximo Asset Management information, complete the following steps:

Procedure

1. Back up the Maximo Asset Management database, J2EE server, and authentication server information by using the instructions provided by your middleware vendors.
2. Create a backup of the installation directory.
By default, this directory is \IBM\SMP.
Ensure that all file permissions are preserved.
3. Back up IBM Installation Manager agent data and shared files for recovery with IBM Business Process Manager (BPM). Refer to the IBM Support site for more information (<http://www-01.ibm.com/support/docview.wss?uid=swg21665878>).

Restoring the administrative server

This section details how to restore previously backed up Maximo Asset Management administrative server information. This information can be used to return an existing Maximo Asset Management administrative server to a previous state.

Before you begin

It is important to restore the backup of the database that was made when you backed up the administrative server. A database backup should be restored with the administrative server backup it was paired with.

About this task

To restore Maximo Asset Management information to an administrative server, complete the following steps:

Procedure

1. Restore the database backup that was paired with the administrative server backup you are restoring.
2. Log on to the target administrative system with the same user ID that was used to install the product on the existing administrative server.
3. Copy the Maximo Asset Management installation files and directories to the file system of the target administrative system.

You must maintain the directory structure of the original installation. For example, if the Maximo Asset Management installation directory on the existing administrative system is `\IBM\SMP`, you cannot copy those files to a `\NewAdminWS\IBM\SMP` directory on the target administrative server.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux[®] is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademark of IBM or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's name, user name, password, or other personally identifiable information for purposes of session management, authentication, single sign-on configuration or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also likely eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's [Privacy Policy](http://www.ibm.com/privacy) at <http://www.ibm.com/privacy> and IBM's [Online Privacy Statement](https://www.ibm.com/privacy/details/us/en/) at <https://www.ibm.com/privacy/details/us/en/> in the section entitled "Cookies, Web Beacons and Other Technologies".

Index

A

- action addLanguages [37](#)
- action configureProducts [38](#)
- action configureProductsDBLite [40](#)
- action deployConfiguration [36](#)
- action deployDatabaseConfiguration [34](#)
- action deployJ2eeConfiguration [35](#)
- action deployMiddlewareConfiguration [38](#)
- action disableAppSecurity [37](#)
- action enableAppSecurity [36](#)
- action enableDB2TextSearch [34](#)
- action reapplyAppServerTuning [36](#)
- action removeConfiguration [36](#)
- action removeDatabaseConfiguration [34](#)
- action removeJ2EEConfiguration [36](#)
- action updateApplication [37](#)
- action updateApplicationDBLite [39](#)
- action updateDatabaseConfiguration [33](#)
- action updateJ2eeConfiguration [35](#)
- action upgrade75Configuration [39](#)
- action validate And Update Database Configuration [33](#)
- action validate And Update J2ee Configuration [35](#)
- action validateDatabaseConfiguration [33](#)
- action validateJ2eeConfiguration [35](#)
- administrative workstation
 - backing up [59](#)
 - restoring [60](#)

B

- backup [59](#)
- best-practice links
 - enabling [16](#)

C

- command line configuration program
 - input properties files [22](#)
 - parameters [23](#)
- Command line configuration program
 - actions [33](#)
- configuration [11](#)
- configuration actions
 - addLanguages [37](#)
 - configureProducts [38](#)
 - configureProductsDBLite [40](#)
 - deployConfiguration [36](#)
 - deployDatabaseConfiguration [34](#)
 - deployJ2eeConfiguration [35](#)
 - deployMiddlewareConfiguration [38](#)
 - disableAppSecurity [37](#)
 - enableAppSecurity [36](#)
 - enableDB2TextSearch [34](#)
 - reapplyAppServerTuning [36](#)
 - removeConfiguration [36](#)
 - removeDatabaseConfiguration [34](#)

- configuration actions (*continued*)
 - removeJ2EEConfiguration [36](#)
 - updateApplication [37](#)
 - updateApplicationDBLite [39](#)
 - updateDatabaseConfiguration [33](#)
 - updateJ2eeConfiguration [35](#)
 - upgrade75Configuration [39](#)
 - validate And Update Database Configuration [33](#)
 - validate And Update J2ee Configuration [35](#)
 - validateDatabaseConfiguration [33](#)
 - validateJ2eeConfiguration [35](#)
- configuring Maximo Asset Management
 - command line configuration program [21](#)
 - validate configuration parameters from the command line [21](#)

E

- EAR files
 - building manually [17](#)
 - maintaining [17](#)

I

- installation
 - program logs [9](#)
- Installation
 - post-installation tasks [11](#)
- installation program
 - logs [9](#)

M

- Maximo Asset Management
 - configuring with the Maximo Asset Management configuration program [1](#)
 - installing version 7.6 [1](#)

N

- NOTF) JMS queue
 - creating [3](#)
- NOTFERR JMS queue
 - creating [5](#)

P

- post installation [11](#)
- properties
 - Maximo Asset Management [40](#)

R

- restoration [59](#)

U

UpdateDB Lite

update database [7](#)

users

groups

synchronizing [14](#)

synchronizing [14](#)

W

WebSphere Application Server Network Deployment

install Maximo Asset Management applications [18](#)

manually uninstalling [18](#)

Work Center

manually building WAR file [17](#)



Part Number:

(1P) P/N: