Installation Reference

Overview

Reference information about Ant tasks and configuration sample files for the installation of IBM MobileFirst Server, IBM MobileFirst Application Center, and IBM MobileFirst Analytics.

Jump to

Ant configuredatabase task reference

Reference information for the configuredatabase Ant task. This reference information is for relational databases only. It does not apply to Cloudant .

The **configuredatabase** Ant task creates the relational databases that are used by MobileFirst Server administration service, MobileFirst Server live update service, MobileFirst Server push service, MobileFirst runtime, and the Application Center services. This Ant task configures a relational database through the following actions:

- Checks whether the MobileFirst tables exist and creates them if necessary.
- If the tables exist for an older version of IBM MobileFirst Foundation, migrates them to the current version.
- If the tables exist for the current version of IBM MobileFirst Foundation, does nothing.

In addition, if one of the following conditions is met:

- The DBMS type is Derby.
- An inner element `<dba>` is present.
- The DBMS type is DB2 , and the specified user has the permissions to create databases.

Then, the task can have the following effects:

- Create the database if necessary (except for Oracle 12c, and Cloudant).
- Create a user, if necessary, and grants that user access rights to the database.

> **Note:** The configuredatabase Ant task has not effect if you use it with Cloudant.

Attributes and elements for configuredatabase task

The **configuredatabase** task has the following attributes:

Attribute	Description	Required	Default
kind	The type of database: In MobileFirst Server: MobileFirstRuntime, MobileFirstConfig, MobileFirstAdmin, or push. In Application Center: ApplicationCenter.	Yes	None
includeConfigurationTables	To specify whether to perform database operations on both the live update service and the administration service or on the administration service only. The value is either true or false.	No	true

Attribute	Description	Required	Default
execute	To specify whether to execute the configuredatabase Ant task. The value is either true or false.	No	true

kind

IBM MobileFirst Foundation V8.0.0 supports four kinds of database: MobileFirst runtime uses **MobileFirstRuntime** database. MobileFirst Server administration service uses the **MobileFirstAdmin** database. MobileFirst Server's Live Update service uses the **MobileFirstConfig** database. By default, it is created with **MobileFirstAdmin** kind. MobileFirst Server push service uses the **push** database. Application Center uses the **ApplicationCenter** database.

includeConfigurationTables

The **includeConfigurationTables** attribute can be used only when the **kind** attribute is **MobileFirstAdmin**. The valid value can be true or false. When this attribute is set to true, the **configuredatabase** task performs database operations on both the administration service database and the Live Update service database in a single run. When this attribute is set to false, the **configuredatabase** task performs database operations only on the administration service database.

execute

The **execute** attribute enables or disables the execution of the **configuredatabase** Ant task. The valid value can be true or false. When this attribute is set to false, the **configuredatabase** task performs no configuration or database operations.

The **configuredatabase** task supports the following elements:

Element	Description	Count
`<derby>`	The parameters for Derby.	0..1
`<db2>`	The parameters for DB2.	0..1
`<mysql>`	The parameters for MySQL.	0..1
`<oracle>`	The parameters for Oracle.	0..1
`<driverclasspath>`	The JDBC driver class path.	0..1

For each database type, you can use a `<property>` element to specify a JDBC connection property for access to the database. The `<property>` element has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value for the property.	Yes	None

Apache Derby

The `<derby>` element has the following attributes:

Attribute	Description	Required	Default
database	The database name.	No	MFPDATA, MFPADM, MFPCFG, MFPPUSH, or APPCNTR, depending on kind.
datadir	The directory that contains the databases.	Yes	None
schema	The schema name.	No	MFPDATA, MFPCFG, MFPADMINISTRATOR, MFPPUSH, or APPCENTER, depending on kind.

The `<derby>` element supports the following element:

Element	Description	Count
`<property>`	The JDBC connection property.	0..∞

For the available properties, see Setting attributes for the database connection URL (http://db.apache.org/derby/docs/10.11/ref/rrefattrib24612.html).

DB2

The `<db2>` element has the following attributes:

Attribute	Description	Required	Default
database	The database name.	No	MFPDATA, MFPADM, MFPCFG, MFPPUSH, or APPCNTR, depending on kind.
server	The host name of the database server.	Yes	None
port	The port on the database server.	No	50000
user	The user name for accessing databases.	Yes	None
password	The password for accessing databases.	No	Queried interactively
instance	The name of the DB2 instance.	No	Depends on the server
schema	The schema name.	No	Depends on the user

For more information about DB2 user accounts, see DB2 security model overview (http://ibm.biz/knowctr#SSEPGG_10.1.0/com.ibm.db2.luw.admin.sec.doc/doc/c0021804.html).
The `<db2>` element supports the following elements:

Element	Description	Count
`<property>`	The JDBC connection property.	0..∞
`<dba>`	The database administrator credentials.	0..1

For the available properties, see Properties for the IBM Data Server Driver for JDBC and SQLJ (http://ibm.biz/knowctr#SSEPGG_10.1.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_rjvdsprp.html).
The inner element `<dba>` specifies the credentials for the database administrators. This element has the following attributes:

Attribute	Description	Required	Default
user	The user name for accessing database.	Yes	None
password	The password or accessing database.	No	Queried interactively

The user that is specified in a `<dba>` element must have the SYSADM or SYSCTRL DB2 privilege. For more information, see Authorities overview (http://ibm.biz/knowctr#SSEPGG_10.1.0/com.ibm.db2.luw.admin.sec.doc/doc/c0055206.html).

The `<driverclasspath>` element must contain the JAR files for the DB2 JDBC driver and for the associated license. You can retrieve those files in one of the following ways:

- Download DB2 JDBC drivers from the DB2 JDBC Driver Versions (http://www.ibm.com/support/docview.wss?uid=swg21363866) page
- Or fetch the **db2jcc4.jar** file and its associated **db2jcc*license**.jar** files from the **DB2*INSTALL*DIR/java** directory on the DB2 server.

You cannot specify details of table allocations, such as the table space, by using the Ant task. To control the table space, you must use the manual instructions in section DB2 database and user requirements (../databases/#db2-database-and-user-requirements).

MySQL

The element `<mysql>` has the following attributes:

Attribute	Description	Required	Default
database	The database name.	No	MFPDATA, MFPADM, MFPCFG, MFPPUSH, or APPCNTR, depending on kind.
server	The host name of the database server.	Yes	None
port	The port on the database server.	No	3306
user	The user name for accessing databases.	Yes	None
password	The password for accessing databases.	No	Queried interactively

For more information about MySQL user accounts, see MySQL User Account Management (http://dev.mysql.com/doc/refman/5.5/en/user-account-management.html).
The `<mysql>` element supports the following elements:

Element	Description	Count
`<property>`	The JDBC connection property.	0..∞
`<dba>`	The database administrator credentials.	0..1
`<client>`	The host that is allowed to access the database.	0..∞

For the available properties, see Driver/Datasource Class Names, URL Syntax and Configuration Properties for Connector/J (http://dev.mysql.com/doc/connector-j/en/connector-j-reference-configuration-properties.html).
The inner element `<dba>` specifies the database administrator credentials. This element has the following attributes:

Attribute	Description	Required	Default
user	The user name for accessing databases.	Yes	None
password	The password for accessing databases.	No	Queried interactively

The user that is specified in a `<dba>` element must be a MySQL superuser account. For more information, see Securing the Initial MySQL Accounts (http://dev.mysql.com/doc/refman/5.5/en/default-privileges.html).

Each `<client>` inner element specifies a client computer or a wildcard for client computers. These computers are allowed to connect to the database. This element has the following attributes:

Attribute	Description	Required	Default
hostname	The symbolic host name, IP address, or template with % as a placeholder.	Yes	None

For more information about the hostname syntax, see Specifying Account Names (http://dev.mysql.com/doc/refman/5.5/en/account-names.html).

The `<driverclasspath>` element must contain a MySQL Connector/J JAR file. You can download that file from the Download Connector/J (http://www.mysql.com/downloads/connector/j/) page.

Alternatively, you can use the `<mysql>` element with the following attributes:

Attribute	Description	Required	Default
url	The database connection URL.	Yes	None
user	The user name for accessing databases.	Yes	None
password	The password for accessing databases.	No	Queried interactively

> `Note:` If you specify the database with the alternative attributes, this database must exist, the user account must exist, and the database must already be accessible to the user. In this case, the **configuredatabase** task does not attempt to create the database or the user, nor does it attempt to grant access to the user. The **configuredatabase** task ensures only that the database has the required tables for the current MobileFirst Server version. You do not have to specify the inner elements `<dba>` or `<client>`.

Oracle

The element `<oracle>` has the following attributes:

Attribute	Description	Required	Default
database	The database name, or Oracle service name. **Note:** You must always use a service name to connect to a PDB database.	No	ORCL
server	The host name of the database server.	Yes	None
port	The port on the database server.	No	1521
user	The user name for accessing databases. See the note under this table.	Yes	None
password	The password for accessing databases.	No	Queried interactively
sysPassword	The password for the user SYS.	No	Queried interactively if the database does not yet exist
systemPassword	The password for the user SYSTEM.	No	Queried interactively if the database or the user does not exist yet

> `Note:` For the user attribute, use preferably a user name in uppercase letters. Oracle user names are generally in uppercase letters. Unlike other database tools, the **configuredatabase** Ant task does not convert lowercase letters to uppercase letters in the user name. If the **configuredatabase** Ant task fails to connect to your database, try to enter the value for the **user** attribute in uppercase letters.

For more information about Oracle user accounts, see Overview of Authentication Methods (http://docs.oracle.com/cd/B28359_01/server.111/b28318/security.htm#i12374).
The `<oracle>` element supports the following elements:

Element	Description	Count
`<property>`	The JDBC connection property.	0..∞
`<dba>`	The database administrator credentials.	0..1

For information about the available connection properties, see Class OracleDriver (http://docs.oracle.com/cd/E11882_01/appdev.112/e13995/oracle/jdbc/OracleDriver.html).
The inner element `<dba>` specifies the database administrator credentials. This element has the following attributes:

Attribute	Description	Required	Default
user	The user name for accessing databases. See the note under this table.	Yes	None
password	The password for accessing databases.	No	Queried interactively

The `<driverclasspath>` element must contain an Oracle JDBC driver JAR file. You can download Oracle JDBC drivers from JDBC, SQLJ, Oracle JPublisher and Universal Connection Pool (UCP) (http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html).

You cannot specify details of table allocation, such as the table space, by using the Ant task. To control the table space, you can create the user account manually and assign it a default table space before you run the Ant task. To control other details, you must use the manual instructions in section Oracle database and user requirements (../databases/#oracle-database-and-user-requirements).

Attribute	Description	Required	Default
url	The database connection URL.	Yes	None
user	The user name for accessing databases.	Yes	None
password	The password for accessing databases.	No	Queried interactively

> **Note:** If you specify the database with the alternative attributes, this database must exist, the user account must exist, and the database must already be accessible to the user. In this case, the task does not attempt to create the database or the user, nor does it attempt to grant access to the user. The **configuredatabase** task ensures only that the database has the required tables for the current MobileFirst Server version. You do not have to specify the inner element `<dba>`.

Ant tasks for installation of MobileFirst Operations Console, MobileFirst Server artifacts, MobileFirst Server administration, and live update services

The **installmobilefirstadmin**, **updatemobilefirstadmin**, and **uninstallmobilefirstadmin** Ant tasks are provided for the installation of MobileFirst Operations Console, the artifacts component, the administration service, and the live update service.

Task effects

installmobilefirstadmin

The **installmobilefirstadmin** Ant task configures an application server to run the WAR files of the administration and live update services as web applications, and optionally, to install the MobileFirst Operations Console. This task has the following effects:

- It declares the administration service web application in the specified context root, by default /mfpadmin.
- It declares the live update service web application in a context root derived from the specified context root of the administration service. By default, /mfpadminconfig.
- For the relational databases, it declares data sources and on WebSphere Application Server full profile, JDBC providers for the administration services.
- It deploys the administration service and the live update service on the application server.
- Optionally, it declaresMobileFirst Operations Console as a web application in the specified context root, by default /mfpconsole. If the MobileFirst Operations Console instance is specified, the Ant task declares the appropriate JNDI environment entry to communicate with the corresponding management service. For example,

```
<target name="adminstall">
 <installmobilefirstadmin servicewar="${mfp.service.war.file}">
  <console install="${mfp.admin.console.install}" warFile="${mfp.console.war.file}"/>
```

- Optionally, it declares the MobileFirst Server artifacts web application in the specified context root /mfp-dev-artifacts

when MobileFirst Operations Console is installed.
- It configures the configuration properties for the administration service by using JNDI environment entries. These JNDI environment entries also give some additional information about the application server topology, for example whether the topology is a stand-alone configuration, a cluster, or a server farm.
- Optionally, it configures users that it maps to roles used by MobileFirst Operations Console, and the administration and live update services web applications.
- It configures the application server for use of JMX.
- Optionally, it configures the communication with the MobileFirst Server push service.
- Optionally, it sets the MobileFirst JNDI environment entries to configure the application server as a server farm member for the MobileFirst Server administration part.

updatemobilefirstadmin

The **updatemobilefirstadmin** Ant task updates an already-configured MobileFirst Server web application on an application server. This task has the following effects:

- It updates the administration service WAR file. This file must have the same base name as the corresponding WAR file that was previously deployed.
- It updates the live update service WAR file. This file must have the same base name as the corresponding WAR file that was previously deployed.
- It updates the MobileFirst Operations Console WAR file. This file must have the same base name as the corresponding WAR file that was previously deployed. The task does not change the application server configuration, that is, the web application configuration, data sources, JNDI environment entries, user-to-role mappings, and JMX configuration.

uninstallmobilefirstadmin

The **uninstallmobilefirstadmin** Ant task undoes the effects of an earlier run of installmobilefirstadmin. This task has the following effects:

- It removes the configuration of the administration service web application with the specified context root. As a consequence, the task also removes the settings that were added manually to that application.
- It removes the WAR files of the administration and live update services, and MobileFirst Operations Console from the application server as an option.
- For the relational DBMS, it removes the data sources and on WebSphere Application Server Full Profile the JDBC providers for the administration and live update services.
- For the relational DBMS, it removes the database drivers that were used by the administration and live update services from the application server.
- It removes the associated JNDI environment entries.
- On WebSphere Application Server Liberty and Apache Tomcat, it removes the users configured by the installmobilefirstadmin invocation.
- It removes the JMX configuration.

Attributes and elements

The **installmobilefirstadmin**, **updatemobilefirstadmin**, and **uninstallmobilefirstadmin** Ant tasks have the following attributes:

Attribute	Description	Required	Default
contextroot	The common prefix for URLs to the administration service to get information about MobileFirst runtime environments, applications, and adapters.	No	/mfpadmin
id	To distinguish different deployments.	No	Empty
environmentId	To distinguish different MobileFirst environments.	No	Empty

Attribute	Description	Required	Default
servicewar	The WAR file for the administration service.	No	The mfp-admin-service.war file is in the same directory as the mfp-ant-deployer.jar file.
shortcutsDir	The directory where to place shortcuts.	No	None
wasStartingWeight	The start order for WebSphere Application Server. Lower values start first.	No	1

contextroot and id

The **contextroot** and **id** attributes distinguish different deployments of MobileFirst Operations Console and the administration service.

In WebSphere Application Server Liberty profiles and in Tomcat environments, the contextroot parameter is sufficient for this purpose. In WebSphere Application Server Full profile environments, the id attribute is used instead. Without this id attribute, two WAR files with the same context roots might conflict and these files would not be deployed.

environmentId

Use the **environmentId** attribute to distinguish several environments, consisting each of MobileFirst Server administration service and MobileFirst runtime web applications, that must operate independently. For example, with this option you can host a test environment, a pre-production environment, and a production environment on the same server or in the same WebSphere Application Server Network Deployment cell. This environmentId attribute creates a suffix that is added to MBean names that the administration service and the MobileFirst runtime projects use when they communicate through Java Management Extensions (JMX).

servicewar

Use the **servicewar** attribute to specify a different directory for the administration service WAR file. You can specify the name of this WAR file with an absolute path or a relative path.

shortcutsDir

The **shortcutsDir** attribute specifies where to place shortcuts to the MobileFirst Operations Console. If you set this attribute, you can add the following files to that directory:

- **mobilefirst-console.url** - this file is a Windows shortcut. It opens the MobileFirst Operations Console in a browser.
- **mobilefirst-console.sh**- this file is a UNIX shell script and opens the MobileFirst Operations Console in a browser.
- **mobilefirst-admin-service.url** - this file is a Windows shortcut. It opens in a browser and calls a REST service that returns a list of the MobileFirst projects that can be managed in JSON format. For each listed MobileFirst project, some details are also available about their artifacts, such as the number of applications, the number of adapters, the number of active devices, the number of decommissioned devices. The list also indicates whether the MobileFirst project runtime is running or idle.
- **mobilefirst-admin-service.sh** - this file is a UNIX shell script that provides the same output as the **mobilefirst-admin-service.url** file.

wasStartingWeight

Use the **wasStartingWeight** attribute to specify a value that is used in WebSphere Application Server as a weight to ensure that a start order is respected. As a result of the start order value, the administration service web application is deployed and started before any other MobileFirst runtime projects. If MobileFirst projects are deployed or started before the web application, the JMX communication is not established and the runtime cannot synchronize with the administration service database and cannot handle server requests.

The **installmobilefirstadmin**, **updatemobilefirstadmin**, and **uninstallmobilefirstadmin** Ant tasks support the following elements:

Element	Description	Count
`<applicationserver>`	The application server.	1
`<configuration>`	The live update service.	1

Element	Description	Count
`<console>`	The administration console.	0..1
`<database>`	The databases.	1
`<jmx>`	To enable Java Management Extensions.	1
`<property>`	The properties.	0..
`<push>`	The push service.	0..1
`<user>`	The user to be mapped to a security role.	0..

To specify a MobileFirst Operations Console

The `<console>` element collects information to customize the installation of the MobileFirst Operations Console. This element has the following attributes:

Attribute	Description	Required	Default
contextroot	The URI of the MobileFirst Operations Console.	No	/mfpconsole
install	To indicate whether the MobileFirst Operations Console must be installed.	No	Yes
warfile	The console WAR file.	No	The mfp-admin-ui.war file is in the same directory as themfp-ant-deployer.jar file.

The `<console>` element supports the following element:

Element	Description	Count
`<artifacts>`	The MobileFirst Server artifacts.	0..1
`<property>`	The properties.	0..

The `<artifacts>` element has the following attributes:

Attribute	Description	Required	Default
install	To indicate whether the artifacts component must be installed.	No	true
warFile	The artifacts WAR file.	No	The mfp-dev-artifacts.war file is in the same directory as the mfp-ant-deployer.jar file

By using this element, you can define your own JNDI properties or override the default value of the JNDI properties that are provided by the administration service and the MobileFirst Operations Console WAR files.

The `<property>` element specifies a deployment property to be defined in the application server. It has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value of the property.	Yes	None

By using this element, you can define your own JNDI properties or override the default value of the JNDI properties that are provided by the administration service and the MobileFirst Operations Console WAR files.

For more information about the JNDI properties, see List of JNDI properties for MobileFirst Server administration service (../server-configuration/#list-of-jndi-properties-for-mobilefirst-server-administration-service).

To specify an application server

Use the `<applicationserver>` element to define the parameters that depend on the underlying application server. The `<applicationserver>` element supports the following elements.

Element	Description	Count
`<websphereapplicationserver>` or `<was>`	The parameters for WebSphere Application Server. The `<websphereapplicationserver>` element (or `was>` in its short form) denotes a WebSphere Application Server instance. WebSphere Application Server full profile (Base, and Network Deployment) are supported, so is WebSphere Application Server Liberty Core and WebSphere Application Server Liberty Network Deployment.	0..1
`<tomcat>`	The parameters for Apache Tomcat.	0..1

The attributes and inner elements of these elements are described in the tables of Ant tasks for installation of MobileFirst runtime environments.
However, for the inner element of the `<was>` element for Liberty collective, see the following table:

Element	Description	Count
`<collectiveController>`	A Liberty collective controller.	0..1

The `<collectiveController>` element has the following attributes:

Attribute	Description	Required	Default
serverName	The name of the collective controller.	Yes	None
controllerAdminName	The administrative user name that is defined in the collective controller. This is the same user that is used to join new members to the collective.	Yes	None
controllerAdminPassword	The administrative user password.	Yes	None
createControllerAdmin	To indicate whether the administrative user must be created in the basic registry of the collective controller. Possible values are true or false.	No	true

To specify the live update service configuration

Use the `<configuration>` element to define the parameters that depend on the live update service. The `<configuration>` element has the following attributes.

Attribute	Description	Required	Default
install	To indicate whether the live update service must be installed.	Yes	true
configAdminUser	The administrator for the live update service.	No. However, it is required for a server farm topology.	If not defined, a user is generated. In a server farm topology, the user name must be the same for all the members of the farm.
configAdminPassword	The administrator password for live update service user.	If a user is specified for **configAdminUser**.	None. In a server farm topology, the password must be the same for all the members of the farm.

Attribute	Description	Required	Default
createConfigAdminUser	To indicate whether to create an admin user in the basic registry of the application server, if it is missing.	No	true
warFile	The live update service WAR file.	No	The mfp-live-update.war file is in the same directory as the mfp-ant-deployer.jar file.

The `<configuration>` element supports the following elements:

Element	Description	Count
`<user>`	The user for the live update service.	0..1
`<property>`	The properties.	0..

The `<user>` element collects the parameters about a user to include in a certain security role for an application.

Attribute	Description	Required	Default
role	A valid security role for the application. Possible value: configadmin.	Yes	None
name	The user name.	Yes	None
password	The password if the user needs to be created.	No	None

After you defined the users by using the `<user>` element, you can map them to any of the following roles for authentication in MobileFirst Operations Console: `configadmin`.

For more information about which authorizations are implied by the specific roles, see Configuring user authentication for MobileFirst Server administration (../server-configuration/#configuring-user-authentication-for-mobilefirst-server-administration).

> **Tip:** If the users exist in an external LDAP directory, set only the **role** and **name** attributes but do not define any passwords.

The `<property>` element specifies a deployment property to be defined in the application server. It has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value of the property.	Yes	None

By using this element, you can define your own JNDI properties or override the default value of the JNDI properties that are provided by the administration service and the MobileFirst Operations Console WAR files. For more information about the JNDI properties, see List of JNDI properties for MobileFirst Server administration service (../server-configuration/#list-of-jndi-properties-for-mobilefirst-server-administration-service).

To specify an application server

Use the `<applicationserver>` element to define the parameters that depend on the underlying application server. The `<applicationserver>` element supports the following elements:

Element	Description	Count
`<webspherapplicationserver>` or `<was>`	The parameters for WebSphere Application Server. The element (or in its short form) denotes a WebSphere Application Server instance. WebSphere Application Server full profile (Base, and Network Deployment) are supported, so is WebSphere Application Server Liberty Core and WebSphere Application Server Liberty Network Deployment.	0..1
`<tomcat>`	The parameters for Apache Tomcat.	0..1

The attributes and inner elements of these elements are described in the tables of Ant tasks for installation of MobileFirst runtime environments.
However, for the inner element of the element for Liberty collective, see the following table:

Element	Description	Count
`<collectiveMember>`	A Liberty collective member.	0..1

The `<collectiveMember>` element has the following attributes:

Attribute	Description	Required	Default
serverName	The name of the collective member.	Yes	None
clusterName	The cluster name that the collective member belongs to.	Yes	None

> **Note:** If the push service and the runtime components are installed in the same collective member, then they must have the same cluster name. If these components are installed on distinct members of the same collective, the cluster names can be different.

To specify Analytics

The `<analytics>` element indicates that you want to connect the MobileFirst push service to an already installed MobileFirst Analytics service. It has the following attributes:

Attribute	Description	Required	Default
install	To indicate whether to connect the push service to MobileFirst Analytics.	No	false
analyticsURL	The URL of MobileFirst Analytics services.	Yes	None
username	The user name.	Yes	None
password	The password.	Yes	None
validate	To validate whether MobileFirst Analytics Console is accessible or not.	No	true

install
Use the install attribute to indicate that this push service must be connected and send events to MobileFirst Analytics. Valid values are true or false.

analyticsURL
Use the analyticsURL attribute to specify the URL that is exposed by MobileFirst Analytics, which receives incoming analytics data.

For example: `http://<hostname>:<port>/analytics-service/rest`

username
Use the username attribute to specify the user name that is used if the data entry point for the MobileFirst Analytics is protected with basic authentication.

password
Use the password attribute to specify the password that is used if the data entry point for the MobileFirst Analytics is protected with basic authentication.

validate
Use the validate attribute to validate whether the MobileFirst Analytics Console is accessible or not, and to check the user name authentication with a password. The possible values are true, or false.

To specify a connection to the push service database

The `<database>` element collects the parameters that specify a data source declaration in an application server to access the push service database.

You must declare a single database: `<database kind="Push">`. You specify the `<database>` element similarly to the configuredatabase Ant task, except that the `<database>` element does not have the `<dba>` and `<client>` elements. It might have `<property>` elements.

The `<database>` element has the following attributes:

Attribute	Description	Required	Default
kind	The kind of database (Push).	Yes	None
validate	To validate whether the database is accessible.	No	true

The `<database>` element supports the following elements. For more information about the configuration of these database elements for relational DBMS, see the tables of Ant tasks for installation of MobileFirst runtime environments.

Element	Description	Count
	The parameter for DB2 databases.	0..1
	The parameter for Apache Derby databases.	0..1
	The parameter for MySQL databases.	0..1
	The parameter for Oracle databases.	0..1
	The parameter for Cloudant databases.	0..1
	The parameter for JDBC driver class path (relational DBMS only).	0..1

> **Note:** The attributes of the `<cloudant>` element are slightly different from the runtime. For more information, see the following table:

Attribute	Description	Required	Default
url	The URL of the Cloudant account.	No	https://user.cloudant.com
user	The user name of the Cloudant account.	Yes	None
password	The password of the Cloudant account.	No	Queried interactively
dbName	The Cloudant database name. **Important:** This database name must start with a lowercase letter and contain only lowercase characters (a-z), Digits (0-9), any of the characters _, $, and -.	No	mfp*push*db

Ant tasks for installation of MobileFirst Server push service

The **installmobilefirstpush**, **updatemobilefirstpush**, and **uninstallmobilefirstpush** Ant tasks are provided for the installation of the push service.

Task effects

installmobilefirstpush

The **installmobilefirstpush** Ant task configures an application server to run the push service WAR file as web application. This task has the following effects: It declares the push service web application in the /**imfpush** context root. The context root cannot be changed. For the relational databases, it declares data sources and, on WebSphere Application Server Full Profile, JDBC providers for push service. It configures the configuration properties for the push service by using JNDI environment entries. These JNDI environment entries configure the OAuth communication with the MobileFirst authorization server, MobileFirst Analytics, and with Cloudant in case Cloudant is used.

updatemobilefirstpush

The **updatemobilefirstpush** Ant task updates an already-configured MobileFirst Server web application on an application server. This task updates the push service WAR file. This file must have the same base name as the corresponding WAR file that was previously deployed.

uninstallmobilefirstpush

The **uninstallmobilefirstpush** Ant task undoes the effects of an earlier run of **installmobilefirstpush**. This task has the following effects: It removes the configuration of the push service web application with the specified context root. As a consequence, the task also removes the settings that were added manually to that application. It removes the push service WAR file from the application server as an option. For the relational DBMS, it removes the data sources and on WebSphere Application Server Full Profile – the JDBC providers for the push service. It removes the associated JNDI environment entries.

Attributes and elements

The **installmobilefirstpush**, **updatemobilefirstpush**, and **uninstallmobilefirstpush** Ant tasks have the following attributes:

Attribute	Description	Required	Default
id	To distinguish different deployments.	No	Empty
warFile	The WAR file for the push service.	No	The ../PushService/mfp-push-service.war file is relative to the MobileFirstServer directory that contains the mfp-ant-deployer.jar file.

Id

The **id** attribute distinguishes different deployments of the push service in the same WebSphere Application Server cell. Without this id attribute, two WAR files with the same context roots might conflict and these files would not be deployed.

warFile

Use the **warFile** attribute to specify a different directory for the push service WAR file. You can specify the name of this WAR file with an absolute path or a relative path.

The **installmobilefirstpush**, **updatemobilefirstpush**, and **uninstallmobilefirstpush** Ant tasks support the following elements:

Element	Description	Count
`<applicationserver>`	The application server.	1
`<analytics>`	The Analytics.	0..1
`<authorization>`	The authorization server for authenticating the communication with other MobileFirst Server components.	1
`<database>`	The databases.	1
`<property>`	The properties.	0..∞

To specify the authorization server

The `<authorization>` element collects information to configure the authorization server for the authentication communication with other MobileFirst Server components. This element has the following attributes:

Attribute	Description	Required	Default
auto	To indicate whether the authorization server URL is computed. The possible values are true or false.	Required on a WebSphere Application Server Network Deployment cluster or node.	true
authorizationURL	The URL of the authorization server.	If mode is not auto.	The context root of the runtime on the local server.
runtimeContextRoot	The context root of the runtime.	No	/mfp
pushClientID	The push service confidential ID in the authorization server.	Yes	None
pushClientSecret	The push service confidential client password in the authorization server.	Yes	None

auto

If the value is set to true, the URL of the authorization server is computed automatically by using the context root of the runtime on the local application server. The auto mode is not supported if you deploy on WebSphere Application Server Network Deployment on a cluster.

authorizationURL

The URL of the authorization server. If the authorization server is the MobileFirst runtime, the URL is the URL of the runtime. For example: `http://myHost:9080/mfp`.

runtimeContextRoot

The context root of the runtime that is used to compute the URL of the authorization server in the automatic mode.

pushClientID

The ID of this push service instance as a confidential client of the authorization server. The ID and the secret must be registered for the authorization server. It can be registered by **installmobilefirstadmin** Ant task, or from MobileFirst Operations Console.

pushClientSecret

The secret key of this push service instance as a confidential client of the authorization server. The ID and the secret must be registered for the authorization server. It can be registered by **installmobilefirstadmin** Ant task, or from MobileFirst Operations Console.

The `<property>` element specifies a deployment property to be defined in the application server. It has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value of the property.	Yes	None

By using this element, you can define your own JNDI properties or override the default value of the JNDI properties that are provided by the push service WAR file.

For more information about the JNDI properties, see List of JNDI properties for MobileFirst Server push service (../server-configuration/#list-of-jndi-properties-for-mobilefirst-server-push-service).

To specify an application server

Use the `<applicationserver>` element to define the parameters that depend on the underlying application server. The `<applicationserver>` element supports the following elements:

Element	Description	Count
or	The parameters for WebSphere Application Server.	The `<websphereapplicationserver>` element (or `<was>` in its short form) denotes a WebSphere Application Server instance. WebSphere Application Server full profile (Base, and Network Deployment) are supported, so is WebSphere Application Server Liberty Core and WebSphere Application Server Liberty Network Deployment.
`<tomcat>`	The parameters for Apache Tomcat.	0..1

The attributes and inner elements of these elements are described in the tables of Ant tasks for installation of MobileFirst runtime environments.

However, for the inner element of the `<was>` element for Liberty collective, see the following table:

Element	Description	Count
`<collectiveMember>`	A Liberty collective member.	0..1

The `<collectiveMember>` element has the following attributes:

Attribute	Description	Required	Default
serverName	The name of the collective member.	Yes	None
clusterName	The cluster name that the collective member belongs to.	Yes	None

> **Note:** If the push service and the runtime components are installed in the same collective member, then they must have the same cluster name. If these components are installed on distinct members of the same collective, the cluster names can be different.

To specify Analytics

The `<analytics>` element indicates that you want to connect the MobileFirst push service to an already installed MobileFirst Analytics service. It has the following attributes:

Attribute	Description	Required	Default
install	To indicate whether to connect the push service to MobileFirst Analytics.	No	false
analyticsURL	The URL of MobileFirst Analytics services.	Yes	None
username	The user name.	Yes	None
password	The password.	Yes	None
validate	To validate whether MobileFirst Analytics Console is accessible or not.	No	true

install

Use the **install** attribute to indicate that this push service must be connected and send events to MobileFirst Analytics. Valid values are true or false.

analyticsURL

Use the **analyticsURL** attribute to specify the URL that is exposed by MobileFirst Analytics, which receives incoming analytics data.
For example: `http://<hostname>:<port>/analytics-service/rest`

username

Use the **username** attribute to specify the user name that is used if the data entry point for the MobileFirst Analytics is protected with basic authentication.

password

Use the **password** attribute to specify the password that is used if the data entry point for the MobileFirst Analytics is protected with basic authentication.

validate

Use the **validate** attribute to validate whether the MobileFirst Analytics Console is accessible or not, and to check the user name authentication with a password. The possible values are true, or false.

To specify a connection to the push service database

The `<database>` element collects the parameters that specify a data source declaration in an application server to access the push service database.

You must declare a single database: `<database kind="Push">`. You specify the `<database>` element similarly to the configuredatabase Ant task, except that the `<database>` element does not have the `<dba>` and `<client>` elements. It might have `<property>` elements.

The `<database>` element has the following attributes:

Attribute	Description	Required	Default
kind	The kind of database (Push).	Yes	None
validate	To validate whether the database is accessible.	No	true

The `<database>` element supports the following elements. For more information about the configuration of these database elements for relational DBMS, see the tables in Ant tasks for installation of MobileFirst runtime environments.

Element	Description	Count
`<db2>`	The parameter for DB2 databases.	0..1
`<derby>`	The parameter for Apache Derby databases.	0..1
`<mysql>`	The parameter for MySQL databases.	0..1
`<oracle>`	The parameter for Oracle databases.	0..1
`<cloudant>`	The parameter for Cloudant databases.	0..1
`<driverclasspath>`	The parameter for JDBC driver class path (relational DBMS only).	0..1

> **Note:** The attributes of the `<cloudant>` element are slightly different from the runtime. For more information, see the following table:

Attribute	Description	Required	Default
url	The URL of the Cloudant account.	No	https://user.cloudant.com
user	The user name of the Cloudant account.	Yes	None
password	The password of the Cloudant account.	No	Queried interactively
dbName	The Cloudant database name. **Important:** This database name must start with a lowercase letter and contain only lowercase characters (a-z), Digits (0-9), any of the characters _, $, and -.	No	mfp*push*db

Ant tasks for installation of MobileFirst runtime environments

Reference information for the **installmobilefirstruntime**, **updatemobilefirstruntime**, and **uninstallmobilefirstruntime** Ant tasks.

Task effects

installmobilefirstruntime

The **installmobilefirstruntime** Ant task configures an application server to run a MobileFirst runtime WAR file as a web application. This task has the following effects.

- It declares the MobileFirst web application in the specified context root, by default /mfp.
- It deploys the runtime WAR file on the application server.
- It declares data sources and on WebSphere Application Server full profile JDBC providers for the runtime.
- It deploys the database drivers in the application server.
- It sets MobileFirst configuration properties through JNDI environment entries.
- Optionally, it sets the MobileFirst JNDI environment entries to configure the application server as a server farm member for the runtime.

updatemobilefirstruntime

The **updatemobilefirstruntime** Ant task updates a MobileFirst runtime that is already configured on an application server. This task updates the runtime WAR file. The file must have the same base name as the runtime WAR file that was previously deployed. Other than that, the task does not change the application server configuration, that is, the web application configuration, data sources, and JNDI environment entries.

uninstallmobilefirstruntime

The **uninstallmobilefirstruntime** Ant task undoes the effects of an earlier **installmobilefirstruntime** run. This task has the following effects.

- It removes the configuration of the MobileFirst web application with the specified context root. The task also removes the settings that are added manually to that application.
- It removes the runtime WAR file from the application server.
- It removes the data sources and on WebSphere Application Server full profile the JDBC providers for the runtime.
- It removes the associated JNDI environment entries.

Attributes and elements

The **installmobilefirstruntime**, **updatemobilefirstruntime**, and **uninstallmobilefirstruntime** Ant tasks have the following attributes:

Attribute	Description	Required	Default
contextroot	The common prefix in URLs to the application (context root).	No	/mfp
id	To distinguish different deployments.	No	Empty
environmentId	To distinguish different MobileFirst environments.	No	Empty
warFile	The WAR file for MobileFirst runtime.	No	The mfp-server.war file is in the same directory as the mfp-ant-deployer.jar file.
wasStartingWeight	The start order for WebSphere Application Server. Lower values start first.	No	2

contextroot and id

The **contextroot** and **id** attributes distinguish different MobileFirst projects.

In WebSphere Application Server Liberty profiles and in Tomcat environments, the contextroot parameter is sufficient for this purpose. In WebSphere Application Server full profile environments, the id attribute is used instead.

environmentId

Use the **environmentId** attribute to distinguish several environments, consisting each of MobileFirst Server administration service and MobileFirst runtime web applications, that must operate independently. You must set this attribute to the same value for the runtime application as the one that was set in the invocation, for the administration service application.

warFile

Use the **warFile** attribute to specify a different directory for the MobileFirst runtime WAR file. You can specify the name of this WAR file with an absolute path or a relative path.

wasStartingWeight

Use the **wasStartingWeight** attribute to specify a value that is used in WebSphere Application Server as a weight to ensure that a start order is respected. As a result of the start order value, the MobileFirst Server administration service web application is deployed and started before any other MobileFirst runtime projects. If MobileFirst projects are deployed or started before the web application, the JMX communication is not established and you cannot manage your MobileFirst projects.

The **installmobilefirstruntime**, **updatemobilefirstruntime**, and **uninstallmobilefirstruntime** tasks support the following elements:

Element	Description	Count
`<property>`	The properties.	0..
`<applicationserver>`	The application server.	1
`<database>`	The databases.	1
`<analytics>`	The analytics.	0..1

The `<property>` element specifies a deployment property to be defined in the application server. It has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value for the property.	Yes	None

The `<applicationserver>` element describes the application server to which the MobileFirst application is deployed. It is a container for one of the following elements:

Element	Description	Count
`<websphereapplicationserver>` or `<was>`	The parameters for WebSphere Application Server.	0..1
`<tomcat>`	The parameters for Apache Tomcat.	0..1

The `<websphereapplicationserver>` element (or `<was>` in its short form) denotes a WebSphere Application Server instance. WebSphere Application Server full profile (Base, and Network Deployment) are supported, so is WebSphere Application Server Liberty Core and WebSphere Application Server Liberty Network Deployment. The `<websphereapplicationserver>` element has the following attributes:

Attribute	Description	Required	Default
installdir	WebSphere Application Server installation directory.	Yes	None
profile	WebSphere Application Server profile, or Liberty.	Yes	None

Attribute	Description	Required	Default
user WebSphere Application Server administrator name.	Yes, except for Liberty	None	
password	WebSphere Application Server administrator password.	No Queried interactively	
libertyEncoding	The algorithm to encode data source passwords for WebSphere Application Server Liberty. The possible values are none, xor, and aes. Whether the xor or aes encoding is used, the clear password is passed as argument to the securityUtility program, which is called through an external process. You can see the password with a ps command, or in the /proc file system on UNIX operating systems.	No	xor
jeeVersion	For Liberty profile. To specify whether to install the features of the JEE6 web profile or the JEE7 web profile. Possible values are 6, 7, or auto.	No	auto
configureFarm	For WebSphere Application Server Liberty, and WebSphere Application Server full profile (not for WebSphere Application Server Network Deployment edition and Liberty collective). To specify whether the server is a server farm member. Possible values are true or false.	No	false
farmServerId	A string that uniquely identify a server in a server farm. The MobileFirst Server administration services and all the MobileFirst runtimes that communicate with it must share the same value.	Yes	None

It supports the following element for single-server deployment:

Element	Description	Count
`<server>`	A single server.	0..1

The element, which is used in this context, has the following attribute:

Attribute	Description	Required	Default
name	The server name.	Yes	None

It supports the following elements for Liberty collective:

Element	Description	Count
`<collectiveMember>`	A Liberty collective member.	0..1

The `<collectiveMember>` element has the following attributes:

Attribute	Description	Required	Default
serverName	The name of the collective member.	Yes	None
clusterName	The cluster name that the collective member belongs to.	Yes	None
serverId	A string that uniquely identifies the collective member.	Yes	None
controllerHost	The name of the collective controller.	Yes	None

Attribute	Description	Required	Default
controllerHttpsPort	The HTTPS port of the collective controller.	Yes	None
controllerAdminName	The administrative user name that is defined in the collective controller. This is the same user that is used to join new members to the collective.	Yes	None
controllerAdminPassword	The administrative user password.	Yes	None
createControllerAdmin	To indicate whether the administrative user must be created in the basic registry of the collective member. Possible values are true or false.	No	true

It supports the following elements for Network Deployment:

Element	Description	Count
`<cell>`	The entire cell.	0..1
`<cluster>`	All the servers of a cluster.	0..1
`<node>`	All the servers in a node, clusters excluded.	0..1
`<server>`	A single server.	0..1

The `<cell>` element has no attributes.

The `<cluster>` element has the following attribute:

Attribute	Description	Required	Default
name	The cluster name.	Yes	None

The `<node>` element has the following attribute:

Attribute	Description	Required	Default
name	The node name.	Yes	None

The `<server>` element, which is used in a Network Deployment context, has the following attributes:

Attribute	Description	Required	Default
nodeName	The node name.	Yes	None
serverName	The server name.	Yes	None

The `<tomcat>` element denotes an Apache Tomcat server. It has the following attribute:

Attribute	Description	Required	Default
installdir	The installation directory of Apache Tomcat. For a Tomcat installation that is split between a CATALINA*HOME directory and a CATALINA*BASE directory, specify the value of the CATALINA_BASE environment variable.	Yes	None
configureFarm	To specify whether the server is a server farm member. Possible values are true or false.	No	false
farmServerId	A string that uniquely identify a server in a server farm. The MobileFirst Server administration services and all the MobileFirst runtimes that communicate with it must share the same value.	Yes	None

The `<database>` element specifies what information is necessary to access a particular database. The `<database>` element is specified like the configuredatabase Ant task, except that it does not have the `<dba>` and `<client>` elements. However, it might have `<property>` elements. The `<database>` element has the following attributes:

Attribute	Description	Required	Default
kind	The kind of database (MobileFirstRuntime).	Yes	None
validate	To validate whether the database is accessible or not. The possible values are true or false.	No	true

The `<database>` element supports the following elements:

Element	Description	Count
`<derby>`	The parameters for Derby.	0..1
`<db2>`	The parameters for DB2.	0..1
`<mysql>`	The parameters for MySQL.	0..1
`<oracle>`	The parameters for Oracle.	0..1
`<driverclasspath>`	The JDBC driver class path.	0..1

The `<analytics>` element indicates that you want to connect the MobileFirst runtime to an already installed MobileFirst Analytics console and services. It has the following attributes:

Attribute	Description	Required	Default
install	To indicate whether to connect the MobileFirst runtime to MobileFirst Analytics.	No	false
analyticsURL	The URL of MobileFirst Analytics services.	Yes	None
consoleURL	The URL ofMobileFirst Analytics Console.	Yes	None
username	The user name.	Yes	None
password	The password.	Yes	None
validate	To validate whether MobileFirst Analytics Console is accessible or not.	No	true
tenant	The tenant for indexing data that is collected from a MobileFirst runtime.	No	Internal identifier

install

Use the **install** attribute to indicate that this MobileFirst runtime must be connected and send events to MobileFirst Analytics. Valid values are **true** or **false**.

analyticsURL

Use the **analyticsURL** attribute to specify the URL that is exposed by MobileFirst Analytics, which receives incoming analytics data.
For example: `http://<hostname>:<port>/analytics-service/rest`

consoleURL

Use the **consoleURL** attribute to the URL that is exposed by MobileFirst Analytics, which links to the MobileFirst Analytics console.
For example: `http://<hostname>:<port>/analytics/console`

username

Use the **username** attribute to specify the user name that is used if the data entry point for the MobileFirst Analytics is protected with basic authentication.

password

Use the **password** attribute to specify the password that is used if the data entry point for the MobileFirst Analytics is protected with basic authentication.

validate

Use the **validate** attribute to validate whether the MobileFirst Analytics Console is accessible or not, and to check the user name authentication with a password. The possible values are **true**, or **false**.

tenant

For more information about this attribute, see Configuration properties (../analytics/configuration/#configuration-properties).

To specify an Apache Derby database

The `<derby>` element has the following attributes:

Attribute	Description	Required	Default
database	The database name.	No	MFPDATA, MFPADM, MFPCFG, MFPPUSH, or APPCNTR, depending on kind.
datadir	The directory that contains the databases.	Yes	None
schema	The schema name.	No	MFPDATA, MFPCFG, MFPADMINISTRATOR, MFPPUSH, or APPCENTER, depending on kind.

The `<derby>` element supports the following element:

Element	Description	Count
`<property>`	The data source property or JDBC connection property.	0..

For more information about the available properties, see the documentation for Class EmbeddedDataSource40 (http://db.apache.org/derby/docs/10.8/publishedapi/jdbc4/org/apache/derby/jdbc/EmbeddedDataSource40.html). See also the documentation for Class EmbeddedConnectionPoolDataSource40 (http://db.apache.org/derby/docs/10.8/publishedapi/jdbc4/org/apache/derby/jdbc/EmbeddedConnectionPoolDataSource40.html).

For more information about the available properties for a Liberty server, see the documentation for `properties.derby.embedded` at Liberty profile: Configuration elements in the server.xml file (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/autodita/rwlp_metatype_4ic.html).

When the **mfp-ant-deployer.jar** file is used within the installation directory of IBM MobileFirst Foundation, a `<driverclasspath>` element is not necessary.

To specify a DB2 database

The `<db2>` element has the following attributes:

Attribute	Description	Required	Default
database	The database name.	No MFPDATA, MFPADM, MFPCFG, MFPPUSH, or APPCNTR, depending on kind.	
server	The host name of the database server.	Yes	None

Attribute	Description	Required	Default
port	The port on the database server.	No	50000
user	The user name for accessing databases.	This user does not need extended privileges on the databases. If you implement restrictions on the database, you can set a user with the restricted privileges	that are listed in Database users and privileges.
password	The password for accessing databases.	No	Queried interactively
schema	The schema name.	No	Depends on the user

For more information about DB2 user accounts, see DB2 security model overview (http://ibm.biz/knowctr#SSEPGG_10.1.0/com.ibm.db2.luw.admin.sec.doc/doc/c0021804.html).
The `<db2>` element supports the following element:

Element	Description	Count
`<property>`	The data source property or JDBC connection property.	0..

For more information about the available properties, see Properties for the IBM Data Server Driver for JDBC and SQLJ (http://ibm.biz/knowctr#SSEPGG_9.7.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_rjvdsprp.html).

For more information about the available properties for a Liberty server, see the **properties.db2.jcc** section at Liberty profile: Configuration elements in the server.xml file (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/autodita/rwlp_metatype_4ic.html).

The `<driverclasspath>` element must contain JAR files for the DB2 JDBC driver and the associated license. You can download DB2 JDBC drivers from DB2 JDBC Driver Versions (http://www.ibm.com/support/docview.wss?uid=swg21363866).

To specify a MySQL database

The `<mysql>` element has the following attributes:

Attribute	Description	Required	Default
database	The database name.	No	MFPDATA, MFPADM, MFPCFG, MFPPUSH, or APPCNTR, depending on kind.
server	The host name of the database server.	Yes	None
port	The port on the database server.	No	3306
user	The user name for accessing databases. This user does not need extended privileges on the databases. If you implement restrictions on the database, you can set a user with the restricted privileges	that are listed in Database users and privileges.	Yes
password	The password for accessing databases.	No	Queried interactively

Instead of **database**, **server**, and **port**, you can also specify a URL. In this case, use the following attributes:

Attribute	Description	Required	Default
url	The URL for connection to the database.	Yes	None

Attribute	Description	Required	Default
user	The user name for accessing databases. This user does not need extended privileges on the databases. If you implement restrictions on the database, you can set a user with the restricted privileges that are listed in Database users and privileges.	Yes	None
password	The password for accessing databases.	No	Queried interactively

For more information about MySQL user accounts, see MySQL User Account Management (http://dev.mysql.com/doc/refman/5.5/en/user-account-management.html).

The `<mysql>` element supports the following element:

Element	Description	Count
`<property>`	The data source property or JDBC connection property.	0..

For more information about the available properties, see the documentation at Driver/Datasource Class Names, URL Syntax and Configuration Properties for Connector/J (http://dev.mysql.com/doc/connector-j/en/connector-j-reference-configuration-properties.html).

For more information about the available properties for a Liberty server, see the properties section at Liberty profile: Configuration elements in the server.xml file (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/autodita/rwlp_metatype_4ic.html).

The `<driverclasspath>` element must contain a MySQL Connector/J JAR file. You can download it from Download Connector/J (http://www.mysql.com/downloads/connector/j/).

To specify an Oracle database

The `<oracle>` element has the following attributes:

Attribute	Description	Required	Default
database	The database name, or Oracle service name. Note: You must always use a service name to connect to a PDB database.	No	ORCL
server	The host name of the database server. Yes None		
port	The port on the database server. No 1521		
user	The user name for accessing databases. This user does not need extended privileges on the databases. If you implement restrictions on the database, you can set a user with the restricted privileges that are listed in Database users and privileges. See the note under this table.	Yes	None
password	The password for accessing databases.	No	Queried interactively

> **Note:** For the **user** attribute, use preferably a user name in uppercase letters. Oracle user names are generally in uppercase letters. Unlike other database tools, the **installmobilefirstruntime** Ant task does not convert lowercase letters to uppercase letters in the user name. If the **installmobilefirstruntime** Ant task fails to connect to your database, try to enter the value for the **user** attribute in uppercase letters.

Instead of **database**, **server**, and **port**, you can also specify a URL. In this case, use the following attributes:

Attribute	Description	Required	Default
url	The URL for connection to the database.	Yes	None

Attribute	Description	Required	Default
user	The user name for accessing databases. This user does not need extended privileges on the databases. If you implement restrictions on the database, you can set a user with the restricted privileges that are listed in Database users and privileges. See the note under this table.	Yes	None
password	The password for accessing databases.	No	Queried interactively

> **Note:** For the **user** attribute, use preferably a user name in uppercase letters. Oracle user names are generally in uppercase letters. Unlike other database tools, the **installmobilefirstruntime** Ant task does not convert lowercase letters to uppercase letters in the user name. If the **installmobilefirstruntime** Ant task fails to connect to your database, try to enter the value for the **user** attribute in uppercase letters.

For more information about Oracle user accounts, see Overview of Authentication Methods (http://docs.oracle.com/cd/B28359_01/server.111/b28318/security.htm#i12374).

For more information about Oracle database connection URLs, see the **Database URLs and Database Specifiers** section at Data Sources and URLs (http://docs.oracle.com/cd/B28359_01/java.111/b31224/urls.htm).

It supports the following element:

Element	Description	Count
`<property>`	The data source property or JDBC connection property.	0..

For more information about the available properties, see the **Data Sources and URLs** section at Data Sources and URLs (http://docs.oracle.com/cd/B28359_01/java.111/b31224/urls.htm).

For more information about the available properties for a Liberty server, see the **properties.oracle** section at Liberty profile: Configuration elements in the server.xml file (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/autodita/rwlp_metatype_4ic.html).

The `<driverclasspath>` element must contain an Oracle JDBC driver JAR file. You can download Oracle JDBC drivers from JDBC, SQLJ, Oracle JPublisher and Universal Connection Pool (UCP) (http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html).

The `<property>` element, which can be used inside `<derby>`, `<db2>`,`<mysql>`, or `<oracle>` elements, has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
type	Java type of the property values, usually java.lang.String/Integer/Boolean.	No	java.lang.String
value	The value for the property.	Yes	None

Ant tasks for installation of Application Center

The `<installApplicationCenter>`, `<updateApplicationCenter>`, and `<uninstallApplicationCenter>` Ant tasks are provided for the installation of the Application Center Console and Services.

Task effects

The `<installApplicationCenter>` task configures an application server to run the Application Center Services WAR file as a web application, and to install the Application Center Console. This task has the following effects:

- It declares the Application Center Services web application in the /applicationcenter context root.
- It declares data sources, and on WebSphere Application Server full profile, it declares also JDBC providers for Application Center Services.

- It deploys the Application Center Services web application on the application server.
- It declares the Application Center Console as a web application in the /appcenterconsole context root.
- It deploys the Application Center Console WAR file on the application server.
- It configures configuration properties for Application Center Services by using JNDI environment entries. The JNDI environment entries that are related to the endpoint and proxies are commented. You must uncomment them in some cases.
- It configures users that it maps to roles used by the Application Center Console and Services web applications.
- On WebSphere Application Server, it configures the necessary custom property for the web container.

The `<updateApplicationCenter>` task updates an already configured Application Center application on an application server. This task has the following effects:

- It updates the Application Center Services WAR file. This file must have the same base name as the corresponding WAR file that was previously deployed.
- It updates the Application Center Console WAR file. This file must have the same base name as the corresponding WAR file that was previously deployed.

The task does not change the application server configuration, that is, the web application configuration, data sources, JNDI environment entries, and user-to-role mappings. This task applies only to an installation that is performed by using the task that is described in this topic.

> **Note:** On WebSphere Application Server Liberty profile, the task does not change the features, which leaves a potential non-minimal list of features in the server.xml file for the installed application.

The `<uninstallApplicationCenter>` Ant task undoes the effects of an earlier run of `<installApplicationCenter>`. This task has the following effects:

- It removes the configuration of the Application Center Services web application with the /**applicationcenter** context root. As a consequence, the task also removes the settings that were added manually to that application.
- It removes both the Application Center Services and Console WAR files from the application server.
- It removes the data sources and, on WebSphere Application Server full profile, it also removes the JDBC providers for the Application Center Services.
- It removes the database drivers that were used by Application Center Services from the application server.
- It removes the associated JNDI environment entries.
- It removes the users who are configured by the `<installApplicationCenter>` invocation.

Attributes and elements

The `<installApplicationCenter>`, `<updateApplicationCenter>`, and `<uninstallApplicationCenter>` tasks have the following attributes:

Attribute	Description	Required	Default
id	It distinguishes different deployments in WebSphere Application Server full profile.	No	Empty
servicewar	The WAR file for the Application Center Services.	No	The applicationcenter.war file is in the application Center console directory: **product*install*dir/ApplicationCenter/console.**
shortcutsDir	The directory where you place the shortcuts.	No	None
aaptDir	The directory that contains the aapt program, from the Android SDK platform-tools package.	No	None

id

In WebSphere Application Server full profile environments, the **id** attribute is used to distinguish different deployments of Application Center Console and Services. Without this **id** attribute, two WAR files with the same context roots might conflict and these files would not be deployed.

servicewar

Use the **servicewar** attribute to specify a different directory for the Application Center Services WAR file. You can specify the name of this WAR file with an absolute path or a relative path.

shortcutsDir

The **shortcutsDir** attribute specifies where to place shortcuts to the Application Center Console. If you set this attribute, the following files are added to this directory:

- **appcenter-console.url**: This file is a Windows shortcut. It opens the Application Center Console in a browser.
- **appcenter-console.sh**: This file is a UNIX shell script. It opens the Application Center Console in a browser.

aaptDir

The **aapt** program is part of the IBM MobileFirst Foundation distribution:
product*install*dir/ApplicationCenter/tools/android-sdk.
If this attribute is not set, during the upload of an apk application, Application Center parses it by using its own code, which might have limitations.

The `<installApplicationCenter>`, `<updateApplicationCenter>`, and `<uninstallApplicationCenter>` tasks support the following elements:

Element	Description	Count
applicationserver	The application server.	1
console	The Application Center console.	1
database	The databases.	1
user	The user to be mapped to a security role.	0..∞

To specify an Application Center console

The `<console>` element collects information to customize the installation of the Application Center Console. This element has the following attributes:

Attribute	Description	Required	Default
warfile	The WAR file for the Application Center Console.	No	The appcenterconsole.war file is in the Application Center console directory: **product*install*dir/ApplicationCenter/console**.

To specify an application server

Use the `<applicationserver>` element to define the parameters that depend on the underlying application server. The `<applicationserver>` element supports the following elements.

Element	Description	Count
websphereapplicationserver or **was**	The parameters for WebSphere Application Server. The `<websphereapplicationserver>` element (or `<was>` in its short form) denotes a WebSphere Application Server instance. WebSphere Application Server full profile (Base, and Network Deployment) are supported, so is WebSphere Application Server Liberty Core. Liberty collective is not supported for Application Center.	0..1
tomcat	The parameters for Apache Tomcat.	0..1

The attributes and inner elements of these elements are described in the tables of the page Ant tasks for installation of MobileFirst runtime environments.

To specify a connection to the services database

The `<database>` element collects the parameters that specify a data source declaration in an application server to access the services database.

You must declare a single database: `<database kind="ApplicationCenter">`. You specify the `<database>` element similarly to the `<configuredatabase>` Ant task, except that the `<database>` element does not have the `<dba>` and `<client>` elements. It might have `<property>` elements.

The `<database>` element has the following attributes:

Attribute	Description	Required	Default
kind	The kind of database (ApplicationCenter).	Yes	None
validate	To validate whether the database is accessible or not.	No	True

The `<database>` element supports the following elements. For more information about the configuration of these database elements, see the tables in Ant tasks for installation of MobileFirst runtime environments.

Element	Description	Count
db2	The parameter for DB2 databases.	0..1
derby	The parameter for Apache Derby databases.	0..1
mysql	The parameter for MySQL databases.	0..1
oracle	The parameter for Oracle databases.	0..1
driverclasspath	The parameter for JDBC driver class path.	0..1

To specify a user and a security role

The `<user>` element collects the parameters about a user to include in a certain security role for an application.

Attribute	Description	Required	Default
role	The user role appcenteradmin.	Yes	None
name	The user name.	Yes	None
password	The password, if you must create the user.	No	None

Ant tasks for installation of MobileFirst Analytics

The **installanalytics**, **updateanalytics**, and **uninstallanalytics** Ant tasks are provided for the installation of MobileFirst Analytics.

The purpose of these Ant Tasks is to configure the MobileFirst Analytics console and the MobileFirst Analytics service with the appropriate storage for the data on an application server. The task installs MobileFirst Analytics nodes that act as a master and data. For more information, see Cluster management and Elasticsearch (../analytics/configuration/#cluster-management-and-elasticsearch).

Task effects

installanalytics

The **installanalytics** Ant task configures an application server to run IBM MobileFirst Analytics. This task has the following effects:

- It deploys the MobileFirst Analytics Service and the MobileFirst Analytics Console WAR files on the application server.
- It declares the MobileFirst Analytics Service web application in the specified context root /analytics-service.
- It declares the MobileFirst Analytics Console web application in the specified context root /analytics.

- It sets MobileFirst Analytics Console and MobileFirst Analytics Services configuration properties through JNDI environment entries.
- On WebSphere Application Server Liberty profile, it configures the web container.
- Optionally, it creates users to use the MobileFirst Analytics Console.

updateanalytics

The **updateanalytics** Ant task updates the already configured MobileFirst Analytics Service and MobileFirst Analytics Console web applications WAR files on an application server. These files must have the same base names as the project WAR files that were previously deployed.

The task does not change the application server configuration, that is, the web application configuration and JNDI environment entries.

uninstallanalytics

The **uninstallanalytics** Ant task undoes the effects of an earlier **installanalytics** run. This task has the following effects:

- It removes the configuration of both the MobileFirst Analytics Service and the MobileFirst Analytics Console web applications with their respective context roots.
- It removes the MobileFirst Analytics Service and the MobileFirst Analytics Console WAR files from the application server.
- It removes the associated JNDI environment entries.

Attributes and elements

The **installanalytics**, **updateanalytics**, and **uninstallanalytics** tasks have the following attributes:

Attribute	Description	Required	Default
serviceWar	The WAR file for the MobileFirst Analytics Service	No	The analytics-service.war file is in the directory Analytics.

serviceWar

Use the **serviceWar** attribute to specify a different directory for the MobileFirst Analytics Services WAR file. You can specify the name of this WAR file with an absolute path or a relative path.

The `<installanalytics>`, `<updateanalytics>`, and `<uninstallanalytics>` tasks support the following elements:

Attribute	Description	Required	Default
console	MobileFirst Analytics	Yes	1
user	The user to be mapped to a security role.	No	0..
storage	The type of storage.	Yes	1
applicationserver	The application server.	Yes	1
property	Properties.	No	0..

To specify a MobileFirst Analytics Console

The `<console>` element collects information to customize the installation of the MobileFirst Analytics Console. This element has the following attributes:

Attribute	Description	Required	Default
warfile	The console WAR file	No	The analytics-ui.war file is in the Analytics directory.
shortcutsdir	The directory where you place the shortcuts.	No	None

warFile

Use the **warFile** attribute to specify a different directory for the MobileFirst Analytics Console WAR file. You can specify the name of this WAR file with an absolute path or a relative path.

shortcutsDir

The **shortcutsDir** attribute specifies where to place shortcuts to the MobileFirst Analytics Console. If you set this attribute, you can add the following files to that directory:

- **analytics-console.url**: This file is a Windows shortcut. It opens the MobileFirst Analytics Console in a browser.
- **analytics-console.sh**: This file is a UNIX shell script. It opens the MobileFirst Analytics Console in a browser.

> Note: These shortcuts do not include the ElasticSearch tenant parameter.

The `<console>` element supports the following nested element:

Element	Description	Count
property	Properties	0..

With this element, you can define your own JNDI properties.

The `<property>` element has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value of the property.	Yes	None

To specify a user and a security role

The `<user>` element collects the parameters about a user to include in a certain security role for an application.

Attribute	Description	Required	Default
role	A valid security role for the application.	Yes	None
name	The user name.	Yes	None
password	The password if the user needs to be created.	No	None

After you defined users by using the`<user>` element, you can map them to any of the following roles for authentication in the MobileFirst Operations Console:

- **mfpmonitor**
- **mfpoperator**
- **mfpdeployer**
- **mfpadmin**

To specify a type of storage for MobileFirst Analytics

The `<storage>` element indicates which underlying type of storage MobileFirst Analytics uses to store the information and data it collects.

It supports the following element:

Element	Description	Count
elasticsearch	ElasticSearch	cluster

The `<elasticsearch>` element collects the parameters about an ElasticSearch cluster.

Attribute	Description	Required	Default
clusterName	The ElasticSearch cluster name.	No	worklight
nodeName	The ElasticSearch node name. This name must be unique in an ElasticSearch cluster.	No	`worklightNode_<random number>`
mastersList	A comma-delimited string that contains the host name and ports of the ElasticSearch master nodes in the ElasticSearch cluster (For example: hostname1:transport-port1,hostname2:transport-port2)	No	Depends on the topology
dataPath	The ElasticSearch cluster location.	No	Depends on the application server
shards	The number of shards that the ElasticSearch cluster creates. This value can be set only by the master nodes that are created in the ElasticSearch cluster.	No	5
replicasPerShard	The number of replicas for each shard in the ElasticSearch cluster. This value can be set only by the master nodes that are created in the ElasticSearch cluster.	No	1
transportPort	The port used for node-to-node communication in the ElasticSearch cluster.	No	9600

clusterName

Use the **clusterName** attribute to specify a name of your choice for the ElasticSearch cluster.

An ElasticSearch cluster consists of one or more nodes that share the same cluster name so you might specify the same value for the **clusterName** attribute if you configure several nodes.

nodeName

Use the **nodeName** attribute to specify a name of your choice for the node to configure in the ElasticSearch cluster. Each node name must be unique in the ElasticSearch cluster even if nodes span on several machines.

mastersList

Use the **mastersList** attribute to provide a comma-separated list of the master nodes in your ElasticSearch cluster. Each master node in this list must be identified by its host name, and the ElasticSearch node-to-node communication port. This port is 9600 by default, or it is the port number that you specified with the attribute **transportPort** when you configured that master node.

For example: `hostname1:transport-port1, hostname2:transport-port2`.

Note:

- If you specify a **transportPort** that is different than the default value 9600, you must also set this value with the attribute **transportPort**. By default, when the attribute **mastersList** is omitted, an attempt is made to detect the host name and the ElasticSearch transport port on all supported application servers.
- If the target application server is WebSphere Application Server Network Deployment cluster, and if you add or remove a server from this cluster at a later point in time, you must edit this list manually to keep in sync with the ElasticSearch cluster.

dataPath

Use the **dataPath** attribute to specify a different directory to store ElasticsSearch data. You can specify an absolute path or a relative path.

If the attribute **dataPath** is not specified, then ElasticSearch cluster data is stored in a default directory that is called **analyticsData**, whose location depends on the application server:

- For WebSphere Application Server Liberty profile, the location is `${wlp.user.dir}/servers/serverName/analyticsData`.
- For Apache Tomcat, the location is `${CATALINA_HOME}/bin/analyticsData`.

- For WebSphere Application Server and WebSphere Application Server Network Deployment, the location is `${was.install.root}/profiles/<profileName>/analyticsData`.

The directory **analyticsData** and the hierarchy of sub-directories and files that it contains are automatically created at run time, if they do not already exist when the MobileFirst Analytics Service component receives events.

shards

Use the **shards** attribute to specify the number of shards to create in the ElasticSearch cluster.

replicasPerShard

Use the **replicasPerShard** attribute to specify the number of replicas to create for each shard in the ElasticSearch cluster.

Each shard can have zero or more replicas. By default, each shard has one replica, but the number of replicas can be changed dynamically on an existing index in the MobileFirst Analytics. A replica shard can never be started on the same node as its shard.

transportPort

Use the **transportPort** attribute to specify a port that other nodes in the ElasticSearch cluster must use when communicating with this node. You must ensure that this port is available and accessible if this node is behind a proxy or firewall.

To specify an application server

Use the `<applicationserver>` element to define the parameters that depend on the underlying application server. The `<applicationserver>` element supports the following elements.

Note: The attributes and inner elements of this element are described in the tables of Ant tasks for installation of MobileFirst runtime environments.

Element	Description	Count
websphereapplicationserver or **was**	The parameters for WebSphere Application Server.	0..1
tomcat	The parameters for Apache Tomcat.	0..1

To specify custom JNDI properties

The `<installanalytics>`, `<updateanalytics>`, and `<uninstallanalytics>` elements support the following element:

Element	Description	Count
property	Properties	0..

By using this element, you can define your own JNDI properties.

This element has the following attributes:

Attribute	Description	Required	Default
name	The name of the property.	Yes	None
value	The value of the property.	Yes	None

Internal runtime databases

Learn about runtime database tables, their purpose, and order of magnitude of data stored in each table. In relational databases, the entities are organized in database tables.

Database used by MobileFirst Server runtime

The following table provides a list of runtime database tables, their descriptions, and how they are used in relational databases.

Relational database table name	Description	Order of magnitude
LICENSE_TERMS	Stores the various license metrics captured every time the device decommissioning task is run.	Tens of rows. This value does not exceed the value set by the JNDI property mfp.device.decommission.when property. For more information about JNDI properties, see List of JNDI properties for MobileFirst runtime (../server-configuration/#list-of-jndi-properties-for-mobilefirst-runtime)
ADDRESSABLE_DEVICE	Stores the addressable device metrics daily. An entry is also added each time that a cluster is started.	About 400 rows. Entries older than 13 months are deleted daily.
MFP*PERSISTENT*DATA	Stores instances of client applications that have registered with the OAuth server, including information about the device, the application, users associated with the client and the device status.	One row per device and application pair.
MFP*PERSISTENT*CUSTOM_ATTR	Custom attributes that are associated with instances of client applications. Custom attributes are application-specific attributes that were registered by the application per each client instance.	Zero or more rows per device and application pair
MFP*TRANSIENT*DATA	Authentication context of clients and devices	Two rows per device and application pair; if using device single sign-on an extra two rows per device. For more information about SSO, see Configuring device single sign-on (SSO) (../../../authentication-and-security/device-sso).
SERVER_VERSION	The product version.	One row

Database used by MobileFirst Server administration service

The following table provides a list of administration database tables, their descriptions, and how they are used in relational databases.

Relational database table name	Description	Order of magnitude
ADMIN_NODE	Stores information about the servers that run the administration service. In a stand-alone topology with only one server, this entity is not used.	One row per server; empty if a stand-alone server is used.
AUDIT_TRAIL	Stores an audit trail of all administrative actions performed with the administration service.	Thousands of rows.
CONFIG_LINKS	Stores the links to the live update service. Adapters and applications might have configurations that are stored in the live update service, and the links are used to find those configurations.	Hundreds of rows. Per adapter, 2-3 rows are used. Per application, 4-6 rows are used.
FARM_CONFIG	Stores the configuration of farm nodes when a server farm is used.	Tens of rows; empty if no server farm is used.

Relational database table name	Description	Order of magnitude
GLOBAL_CONFIG	Stores some global configuration data.	1 row.
PROJECT	Stores the names of the deployed projects.	Tens of rows.
PROJECT_LOCK	Internal cluster synchronization tasks.	Tens of rows.
TRANSACTIONS	Internal cluster synchronization table; stores the state of all current administrative actions.	Tens of rows.
MFPADMIN_VERSION	The product version.	One row.

Database used by MobileFirst Server live update service

The following table provides a list of live update service database tables, their descriptions, and how they are used in relational databases.

Relational database table name	Description	Order of magnitude
CS_SCHEMAS	Stores the versioned schemas that exist in the platform.	One row per schema.
CS_CONFIGURATIONS	Stores instances of configurations for each versioned schema.	One row per configuration
CS_TAGS	Stores the searchable fields and values for each configuration instance.	Row for each field name and value for each searchable field in configuration.
CS_ATTACHMENTS	Stores the attachments for each configuration instance.	One row per attachment.
CS_VERSION	Stores the version of the MFP that created the tables or instances.	Single row in the table with the version of MFP.

Database used by MobileFirst Server push service

The following table provides a list of push service database tables, their descriptions, and how they are used in relational databases.

Relational database table name	Description	Order of magnitude
PUSH_APPS	Push notification table; stores details of push applications.	One row per application.
PUSH_ENV	Push notification table; stores details of push environments.	Tens of rows.
PUSH_TAGS	Push notification table; stores details of defined tags.	Tens of rows.
PUSH_DEVICES	Push notification table. Stores a record per device.	One row per device.
PUSH_SUBSCRIPTIONS	Push notification table. Stores a record per tag subscription.	One row per device subscription.
PUSH_MESSAGES	Push notification table; stores details of push messages.	Tens of rows.

Relational database table name	Description	Order of magnitude
PUSH*MESSAGE*SEQUENCE_TABLE	Push notification table; stores the generated sequence ID.	One row.
PUSH_VERSION	The product version.	One row.

For more information about setting up the databases, see Setting up databases (../databases).

Sample configuration files

IBM MobileFirst Foundation includes a number of sample configuration files to help you get started with the Ant tasks to install the MobileFirst Server.

The easiest way to get started with these Ant tasks is by working with the sample configuration files provided in the **MobileFirstServer/configuration-samples/** directory of the MobileFirst Server distribution. For more information about installing MobileFirst Server with Ant tasks, see Installing with Ant Tasks (../appserver/#installing-with-ant-tasks).

List of sample configuration files

Pick the appropriate sample configuration file. The following files are provided.

Task	Derby	DB2	MySQL	Oracle
Create databases with database administrator credentials	create-database-derby.xml	create-database-db2.xml	create-database-mysql.xml	create-database-oracle.xml
Install MobileFirst Server on Liberty	configure-liberty-derby.xml	configure-liberty-db2.xml	configure-liberty-mysql.xml	(See Note on MySQL)
Install MobileFirst Server on WebSphere Application Server full profile, single server	configure-was-derby.xml	configure-was-db2.xml	configure-was-mysql.xml (See Note on MySQL)	configure-was-oracle.xml
Install MobileFirst Server on WebSphere Application Server Network Deployment (See Note on configuration files)	configure-wasnd-cluster-derby.xml, configure-wasnd-server-derby.xml, configure-wasnd-node-derby.xml. configure-wasnd-cell-derby.xml	configure-wasnd-cluster-db2.xml, configure-wasnd-server-db2.xml, configure-wasnd-node-db2.xml, configure-wasnd-cell-db2.xml	configure-wasnd-cluster-mysql.xml (See Note on MySQL), configure-wasnd-server-mysql.xml (See Note on MySQL), configure-wasnd-node-mysql.xml (See Note on MySQL), configure-wasnd-cell-mysql.xml	configure-wasnd-cluster-oracle.xml, configure-wasnd-server-oracle.xml, configure-wasnd-node-oracle.xml, configure-wasnd-cell-oracle.xml
Install MobileFirst Server on Apache Tomcat	configure-tomcat-derby.xml	configure-tomcat-db2.xml	configure-tomcat-mysql.xml	configure-tomcat-oracle.xml

Task	Derby	DB2	MySQL	Oracle
Install MobileFirst Server on Liberty collective	Not relevant	configure-libertycollective-db2.xml	configure-libertycollective-mysql.xml	configure-libertycollective-oracle.xml

Note on MySQL: MySQL in combination with WebSphere Application Server Liberty profile or WebSphere Application Server full profile is not classified as a supported configuration. For more information, see WebSphere Application Server Support Statement. Consider using IBM DB2 or another database that is supported by WebSphere Application Server to benefit from a configuration that is fully supported by IBM Support.

Note on configuration files for WebSphere Application Server Network Deployment: The configuration files for **wasnd** contain a scope that can be set to **cluster**, **node**, **server**, or **cell**. For example, for **configure-wasnd-cluster-derby.xml**, the scope is **cluster**. These scope types define the deployment target as follows:

- **cluster**: To deploy to a cluster.
- **server**: To deploy to a single server that is managed by the deployment manager.
- **node**: To deploy to all the servers that are running on a node, but that do not belong to a cluster.
- **cell**: To deploy to all the servers on a cell.

Sample configuration files for MobileFirst Analytics

BM MobileFirst Foundation includes a number of sample configuration files to help you get started with the Ant tasks to install the MobileFirst Analytics Services, and the MobileFirst Analytics Console.

The easiest way to get started with the `<installanalytics>`, `<updateanalytics>`, and `<uninstallanalytics>` Ant tasks is by working with the sample configuration files provided in the **Analytics/configuration-samples/** directory of the MobileFirst Server distribution.

Step 1

Pick the appropriate sample configuration file. The following XML files are provided. They are referred to as **configure-file.xml** in the next steps.

Task	Application server
Install MobileFirst Analytics Services and Console on WebSphere Application Server Liberty profile	configure-liberty-analytics.xml
Install MobileFirst Analytics Services and Console on Apache Tomcat	configure-tomcat-analytics.xml
Install MobileFirst Analytics Services and Console on WebSphere Application Server full profile	configure-was-analytics.xml
Install MobileFirst Analytics Services and Console on WebSphere Application Server Network Deployment, single server	configure-wasnd-server-analytics.xml
Install MobileFirst Analytics Services and Console on WebSphere Application Server Network Deployment, cell	configure-wasnd-cell-analytics.xml
Install MobileFirst Analytics Services and Console on WebSphere Application Server Network Deployment, node	configure-wasnd-node.xml
Install MobileFirst Analytics Services and Console on WebSphere Application Server Network Deployment, cluster	configure-wasnd-cluster-analytics.xml

Note on configuration files for WebSphere Application Server Network Deployment:
The configuration files for wasnd contain a scope that can be set to **cluster**, **node**, **server**, or **cell**. For example, for **configure-wasnd-cluster-analytics.xml**, the scope is **cluster**. These scope types define the deployment target as follows:

- **cluster**: To deploy to a cluster.

- **server**: To deploy to a single server that is managed by the deployment manager.
- **node**: To deploy to all the servers that are running on a node, but that do not belong to a cluster.
- **cell**: To deploy to all the servers on a cell.

Step 2

Change the file access rights of the sample file to be as restrictive as possible. Step 3 requires that you supply some passwords. If you must prevent other users on the same computer from learning these passwords, you must remove the read permissions of the file for users other than yourself. You can use a command, such as the following examples:

On UNIX: `chmod 600 configure-file.xml` On Windows: `cacls configure-file.xml /P Administrators:F %USERDOMAIN%\%USERNAME%:F`

Step 3

Similarly, if your application server is WebSphere Application Server Liberty profile, or Apache Tomcat, and the server is meant to be started only from your user account, you must also remove the read permissions for users other than yourself from the following files:

- For WebSphere Application Server Liberty profile: **wlp/usr/servers**//**server.xml**
- For Apache Tomcat: **conf/server.xml**

Step 4

Replace the placeholder values for the properties at the beginning of the file.

Note:
The following special characters must be escaped when they are used in the values of the Ant XML scripts:

- The dollar sign (`$`) must be written as $$, unless you explicitly want to reference an Ant variable through the syntax `${variable}`, as described in Properties section of the Apache Ant Manual.
- The ampersand character (`&`) must be written as `&amp;`, unless you explicitly want to reference an XML entity.
- Double quotation marks (`"`) must be written as `&quot;`, except when it is inside a string that is enclosed in single quotation marks.

Step 5

Run the command: `ant -f configure-file.xml install`

This command installs your MobileFirst Analytics Services and MobileFirst Analytics Console components in the application server. To install updated MobileFirst Analytics Services and MobileFirst Analytics Console components, for example if you apply a MobileFirst Server fix pack, run the following command: `ant -f configure-file.xml minimal-update`.

To reverse the installation step, run the command: `ant -f configure-file.xml uninstall`

This command uninstalls the MobileFirst Analytics Services and MobileFirst Analytics Console components.

Last modified on