

Command-line tool for uploading or deleting an application

Overview

To deploy applications to the Application Center through a build process, use the command-line tool.

You can upload an application to the Application Center by using the web interface of the Application Center console. You can also upload a new application by using a command-line tool.

This is particularly useful when you want to incorporate the deployment of an application to the Application Center into a build process. This tool is located at:

installDir/ApplicationCenter/tools/applicationcenterdeploytool.jar.

The tool can be used for application files with extension APK or IPA. It can be used stand alone or as an ant task.

The tools directory contains all the files required to support the use of the tool.

- **applicationcenterdeploytool.jar**: the upload tool.
- **json4j.jar**: the library for the JSON format required by the upload tool.
- **build.xml**: a sample ant script that you can use to upload a single file or a sequence of files to the Application Center.
- **acdeploytool.sh** and **acdeploytool.bat**: Simple scripts to call java with **applicationcenterdeploytool.jar**.

Jump to

- Using the stand-alone tool to upload an application
- Using the stand-alone tool to delete an application
- Using the stand-alone tool to clear the LDAP cache
- Ant task for uploading or deleting an application

Using the stand-alone tool to upload an application

To upload an application, call the stand-alone tool from the command line.

Use the stand-alone tool by following these steps.

1. Add **applicationcenterdeploytool.jar** and **json4j.jar** to the java classpath environment variable.
2. Call the upload tool from the command line:

```
java com.ibm.appcenter.Upload [options] [files]
```

You can pass any of the available options in the command line.

Option	Content	
	indicated by	Description
-s	serverpath	The path to the Application Center server.
-c	context	The context of the Application Center web application.
-u	user	The user credentials to access the Application Center.

Option	Content indicated by	Description
-p	password	The password of the user.
-d	description	The description of the application to be uploaded.
-l	label	The fallback label. Normally the label is taken from the application descriptor stored in the file to be uploaded. If the application descriptor does not contain a label, the fallback label is used.
-isActive	true or false	The application is stored in the Application Center as an active or inactive application.
-isInstaller	true or false	The application is stored in the Application Center with the “installer” flag set appropriately.
-isReadyForProduction	true or false	The application is stored in the Application Center with the “ready-for-production” flag set appropriately.
-isRecommended	true or false	The application is stored in the Application Center with the “recommended” flag set appropriately.
-e		Shows the full exception stack trace on failure.
-f		Force uploading of applications, even if they exist already.
-y		Disable SSL security checking, which allows publishing on secured hosts without verification of the SSL certificate.

The files parameter can specify files of type Android application package (.apk) files or iOS application (.ipa) files.

In this example user demo has the password demopassword. Use this command line.

```
java com.ibm.appcenter.Upload -s http://localhost:9080 -c applicationcenter -u demo -p demopassword -f app1.ipa app2.ipa
```

Using the stand-alone tool to delete an application

To delete an application from the Application Center, call the stand-alone tool from the command line. Use the stand-alone tool by following these steps.

1. Add **applicationcenterdeploytool.jar** and **json4j.jar** to the java classpath environment variable.
2. Call the upload tool from the command line:

```
java com.ibm.appcenter.Upload -delete [options] [files or applications]
```

You can pass any of the available options in the command line.

Option	Content indicated by	Description
--------	----------------------	-------------

-s	serverpath	The path to the Application Center server.
-c	context	The context of the Application Center web application.
-u	user	The user credentials to access the Application Center.
-p	password	The password of the user.
-y		Disable SSL security checking, which allows publishing on secured hosts without verification of the SSL certificate. Use of this flag is a security risk, but may be suitable for testing localhost with temporary self-signed SSL certificates.

You can specify files or the application package, operating system, and version. If files are specified, the package, operating system and version are determined from the file and the corresponding application is deleted from the Application Center. If applications are specified, they must have one of the following formats:

- `package@os@version`: This exact version is deleted from the Application Center. The version part must specify the “internal version”, not the “commercial version” of the application.
- `package@os`: All versions of this application are deleted from the Application Center.
- `package`: All versions of all operating systems of this application are deleted from the Application Center.

Example

In this example, user demo has the password demopassword. Use this command line to delete the iOS application demo.HelloWorld with internal version 3.0.

```
java com.ibm.appcenter.Upload -delete -s http://localhost:9080 -c applicationcenter -u demo -p demopass
word demo.HelloWorld@iOS@3.0
```

Using the stand-alone tool to clear the LDAP cache

Use the stand-alone tool to clear the LDAP cache and make changes to LDAP users and groups visible immediately in the Application Center.

When the Application Center is configured with LDAP, changes to users and groups on the LDAP server become visible to the Application Center after a delay. The Application Center maintains a cache of LDAP data and the changes only become visible after the cache expires. By default, the delay is 24 hours. If you do not want to wait for this delay to expire after changes to users or groups, you can call the stand-alone tool from the command line to clear the cache of LDAP data. By using the stand-alone tool to clear the cache, the changes become visible immediately.

Use the stand-alone tool by following these steps.

1. Add applicationcenterdeploytool.jar and json4j.jar to the java classpath environment variable.
2. Call the upload tool from the command line:

```
java com.ibm.appcenter.Upload -clearLdapCache [options]
```

You can pass any of the available options in the command line.

Content

Optionindicated Description

by

-s	serverpath	The path to the Application Center server.
-c	context	The context of the Application Center web application.
-u	user	The user credentials to access the Application Center.
-p	password	The password of the user.
-y		Disable SSL security checking, which allows publishing on secured hosts without verification of the SSL certificate. Use of this flag is a security risk, but may be suitable for testing localhost with temporary self-signed SSL certificates.

Example

In this example, user demo has the password demopassword.

```
java com.ibm.appcenter.Upload -clearLdapCache -s http://localhost:9080 -c applicationcenter -u demo -p
demopassword
```

Ant task for uploading or deleting an application

You can use the upload and delete tools as an Ant task and use the Ant task in your own Ant script. Apache Ant is required to run these tasks. The minimum supported version of Apache Ant is listed in System requirements (../product-overview/requirements).

For convenience, Apache Ant 1.8.4 is included in IBM MobileFirst Foundation Server. In the product *install*/dir/shortcuts/ directory, the following scripts are provided:

- ant for UNIX / Linux
- ant.bat for Windows

These scripts are ready to run, which means that they do not require specific environment variables. If the environment variable JAVA_HOME is set, the scripts accept it.

When you use the upload tool as an Ant task, the classname value of the upload Ant task is **com.ibm.appcenter.ant.UploadApps**. The classname value of the delete Ant task is **com.ibm.appcenter.ant.DeleteApps**.

Parameters of Ant task	Description
serverPath	To connect to the Application Center. The default value is http://localhost:9080.
context	The context of the Application Center. The default value is /applicationcenter.
loginUser	The user name with permissions to upload an application.
loginPass	The password of the user with permissions to upload an application.
forceOverwrite	If this parameter is set to true, the Ant task attempts to overwrite applications in the Application Center when it uploads an application that is already present. This parameter is available only in the upload Ant task.
file	The .apk or .ipa file to be uploaded to the Application Center or to be deleted from the Application Center. This parameter has no default value.
fileset	To upload or delete multiple files.
application	The package name of the application; this parameter is available only in the delete Ant task.
os	The operating system of the application. (For example, Android or iOS.) This parameter is available only in the delete Ant task.
version	The internal version of the application; this parameter is available only in the delete Ant task. Do not use the commercial version here, because the commercial version is unsuitable to identify the version exactly.

Example

You can find an extended example in the **ApplicationCenter/tools/build.xml** directory. The following example shows how to use the Ant task in your own Ant script.

```

<?xml version="1.0" encoding="UTF-8"?>
<project name="PureMeapAntDeployTask" basedir="." default="upload.AllApps">

  <property name="install.dir" value="../../" />
  <property name="workspace.root" value="../../" />

  <!-- Server Properties -->
  <property name="server.path" value="http://localhost:9080/" />
  <property name="context.path" value="applicationcenter" />
  <property name="upload.file" value="" />
  <property name="force" value="true" />

  <!-- Authentication Properties -->
  <property name="login.user" value="appcenteradmin" />
  <property name="login.pass" value="admin" />
  <path id="classpath.run">
    <fileset dir="${install.dir}/ApplicationCenter/tools/">
      <include name="applicationcenterdeploytool.jar" />
      <include name="json4j.jar"/>
    </fileset>
  </path>
  <target name="upload.init">
    <taskdef name="uploadapps" classname="com.ibm.appcenter.ant.UploadApps">
      <classpath refid="classpath.run" />
    </taskdef>
  </target>
  <target name="upload.App" description="Uploads a single application" depends="upload.init">
    <uploadapps serverPath="${server.path}"
      context="${context.path}"
      loginUser="${login.user}"
      loginPass="${login.pass}"
      forceOverwrite="${force}"
      file="${upload.file}" />
  </target>
  <target name="upload.AllApps" description="Uploads all found APK and IPA files" depends="upload.init">
    <uploadapps serverPath="${server.path}"
      loginUser="${login.user}"
      loginPass="${login.pass}"
      forceOverwrite="${force}"
      context="${context.path}" >
      <fileset dir="${workspace.root}">
        <include name="**/*.ipa" />
      </fileset>
    </uploadapps>
  </target>
</project>

```

This sample Ant script is in the **tools** directory. You can use it to upload a single application to the Application Center.

```
ant upload.App -Dupload.file=sample.ipa
```

You can also use it to upload all applications that are found in a directory hierarchy.

```
ant upload.AllApps -Dworkspace.root=myDirectory
```

Properties of the sample Ant script

Property	Comment
install.dir	Defaults to ../../
server.path	The default value is http://localhost:9080.
context.path	The default value is applicationcenter.
upload.file	This property has no default value. It must include the exact file path.
workspace.root	Defaults to ../../
login.user	The default value is appcenteradmin.
login.pass	The default value is admin.
force	The default value is true.

To specify these parameters by command line when you call Ant, add -D before the property name. For example:

```
-Dserver.path=http://localhost:8888/
```