

# Implementing Secure Direct Update

## Overview

For secure Direct Update to work, a user-defined keystore file must be deployed in MobileFirst Server and a copy of the matching public key must be included in the deployed client application.

This topic describes how to bind a public key to new client applications and existing client applications that were upgraded. For more information on configuring the keystore in MobileFirst Server, see [Configuring the MobileFirst Server keystore \(../authentication-and-security/configuring-the-mobilefirst-server-keystore/\)](#).

The server provides a built-in keystore that can be used for testing secure Direct Update for development phases.

**Note:** After you bind the public key to the client application and rebuild it, you do not need to upload it again to the MobileFirst Server. However, if you previously published the application to the market, without the public key, you must republish it.

For development purposes, the following default, dummy public key is provided with MobileFirst Server:

```
-----BEGIN PUBLIC KEY-----
MIIDPjCCAiagAwIBAgIEUD3/bjANBgkqhkiG9w0BAQsFADBGMQswCQYDVQQGEwJJTDELMAkGA1UEC
BMCSUwxETA
PBgNVBAcTCFNoZWZheWltMQwwCgYDVQQKEwNJQk0xEjAQBgNVBAcTCVdvcmtsaWdodDEPMA0GA
1UEAxMGMGV0wgRG
V2MCAXDTEyMDgyOTExMzkyNloYDzQ3NTAwNzI3MTEzOTI2WjBgMQswCQYDVQQGEwJJTDELMAkG
A1UECBMCSUwxETA
TAPBgNVBAcTCFNoZWZheWltMQwwCgYDVQQKEwNJQk0xEjAQBgNVBAcTCVdvcmtsaWdodDEPMA0
GA1UEAxMGMGV0wg
RGV2MIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzQN3vEB2/of7KAuvyolt0T7cjaSTjnOB
m0N3+q
zx++dh92KpNJXj/a3o4YbwJXk7jU8ykjCYvjXRf0hme+HGhilVwxJo54iqh76skDS5m7DaseFdndZUJ4p7NF
Vw
I5ixA36ZArSZ/Pn/ej56/RRjBeRI7AEGXUSGojBUPA6J6DYkwaXQRew9I+Q1kj4dTigyKL5Os0vNFaQyYu+
bT2E
vnOixQ0DXm94lqmHZamZKbZLrWcOEfuAsSjKYOdMSM9jkCiHaKcj7fpEZhUxRRs7joKs1Ri4ihs6JeUvME
iG4gK
I9V3FP/Huy0pfkL0F8xMHgaQ4c/lxS/s3PV0OEg+7wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAQAgEhh
qRI2Rgkt
MJeQOCRCt3uyr4XDK3hmuhEaE0nOvLHi61PoLKnDUNryWUicK/W+tUP9jkN5xRckdzG6TJ/HPySmZ7Ad
r6QRFu+
xcIMY+/S8j4PHLXBjoqgtUMhkt7S2/thN/VA6mwZpw4OI0Pa2hyT2TkhQoYYkRwYCK9pxmuBCoH/eCWpS
xquNny
RwrY25x0YzccXUaMI8L3/3hzq3mW40YIMiEdpiD5HqjUDpzN1funHNQdsxEIMYsWmGAwOdV5sIFzyrH+E
rUYUFA
pdGldLtkrhzbqHFwXE0v3dt+lnLf21wRPIqYHaEu+EB/A4dLO6hm+lJBeu/No7H7TBFm
-----END PUBLIC KEY-----
```

Important: Do not use the public key for production purposes.

## Generating and deploying the keystore

There are many tools available for generating certificates and extracting public keys from a keystore. The following example demonstrates the procedures with the JDK keytool utility and openSSL.

1. Extract the public key from the keystore file that is deployed in the MobileFirst Server.

Note: The public key must be Base64 encoded.

For example, assume that the alias name is `mfp-server` and the keystore file is **keystore.jks**.

To generate a certificate, issue the following command:

```
keytool -export -alias mfp-server -file certfile.cert  
-keystore keystore.jks -storepass keypassword
```

A certificate file is generated.

Issue the following command to extract the public key:

```
openssl x509 -inform der -in certfile.cert -pubkey -noout
```

**Note:** Keytool alone cannot extract public keys in Base64 format.

2. Perform one of the following procedures:
  - Copy the resulting text, without the `BEGIN PUBLIC KEY` and `END PUBLIC KEY` markers into the `mfpclient` property file of the application, immediately after `wlSecureDirectUpdatePublicKey`.
  - From the command prompt, issue the following command: `mfpdev app config direct_update_authenticity_public_key <public_key>`

For `<public_key>`, paste the text that results from Step 1, without the `BEGIN PUBLIC KEY` and `END PUBLIC KEY` markers.

3. Run the cordova build command to save the public key in the application.