Modern Client Management and BigFix Mobile

Installation and Configuration Guide

Special notice

Before using this information and the product it supports, read the information in Notices (on page lxiii).

Edition notice

This edition applies to BigFix version 10 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Audience

This guide is for administrators and IT managers who want to install and configure BigFix MCM and BigFix Mobile. It details prerequisites for each of the scenario and provides installation instructions that allow you to deploy the program in your environment. It also includes information about configuring and maintaining BigFix MCM and BigFix Mobile components.

Chapter 2. Planning the installation

Read this section before you begin to install or update any of the BigFix MCM or BigFix Mobile product features. Effective planning and an understanding of the key aspects of the installation process can help ensure a successful installation.

The BigFix MCM and BigFix Mobile solution is composed of the following BigFix infrastructure components:

- BigFix Platform
    - Enterprise Server
    - BigFix WebUI
    - BigFix DMZ Relay
- BigFix PlugIn Portal
    - BigFix PlugIn for Windows
    - BigFix PlugIn for Apple
    - BigFix PugIn for Android
- BigFix MDM server
    - Windows MDM service
    - Apple MDM service
    - Android MDM service

The BigFix PlugIn Portal and BigFix MDM server will have one or more Plugin and MDM services configured based on the product license entitlements and/or use cases to be exercised; Example, a BigFix Mobile deployment is only entitled to deploy the Android and Apple BigFix Plugin and MDM services.

The following sections in this guide assume that the BigFix Platform components are already installed. For details on installing BigFix and its components, see BigFix Installation Guide.

Chapter 3. On-premises deployment setup

Understand the prerequisites and preparation required to install the BigFix MDM server and BigFix PlugIn Portal on-premise.

Tip: You can perform installation tasks through BigFix WebUI.

Related information

Prerequisites and requirements *(on page 8)*

BigFix PlugIn and MDM SSL certificates and keys *(on page 11)*

LDAPS authentication

Apple Push Notification certificates *(on page 12)*

WNS credentials *(on page 13)*

Google Enterprise Credentials *(on page 13)*

Installing BigFix PlugIn Portal *(on page 13)*

Installing MDM services *(on page 14)*

Installing MDM Plugin *(on page 15)*

Prerequisites and requirements

The following packages must be pre-installed on your Red Hat® Enterprise Linux® systems before you install BigFix MCM and BigFix Mobile:

BigFix MDM server

The target computers must have the following elements installed:

- The computer must be running on RHEL 7+ or RHEL 8+
- Docker (CE v19.x or RHEL version 1.13 or later) and Docker Compose 1.25.x
- BigFix client version 10.0.2 or later (recommended version 10.0.8)
- OpenSSL

Important:

Up to MCM 2.1, port 5671 uses TLS 1.0 for internal communication. If the vulnerability scan detects exposure, you can ignore it. For more information, see Ignore MDM server vulnerability due to TLS 1.0 (on page 61).

Note: RHEL8 distribution no longer provides Docker CE. To install a compatible Docker CE version on RHEL8, see Installing Docker CE and Docker compose on RHEL8 *(on page 46)*.

The BigFix MDM server is typically deployed in the DMZ. Hence, appropriate security measures must be applied to the OS, firewall configuration, and system accounts.

BigFix PlugIn Portal

The target computers must have the following elements installed:

- BigFix Client version 10.0.2 or later (recommended version 10.0.8)
- BigFix PlugIn Portal version 10.0.2 or later (recommended version 10.0.8)

Related information

Minimum hardware requirements *(on page 9)*

TCP/IP Port requirements *(on page 9)*

Supported system environments *(on page 41)*

Minimum hardware requirements

For minimum hardware requirements, see the BigFix Capacity Planning documentation.

For further details and latest information about deployment and management of BigFix, see BigFix Performance & Capacity Planning Resources.

TCP/IP Port requirements

For BigFix MDM Server and BigFix PlugIn Portal to communicate properly with the devices that you manage, the following TCP/IP ports are required.

Port Number	Type	Purpose	Direction
443	HTTPS	All device enrollment and management requests are sent to this port. This must be an internet-facing port for the endpoints to reach the enrollment server.	Inbound to the MDM Server from the network where MDM managed end-points are located.
443	HTTPS	MDM Server to Offline Domain Join Server	Inbound to the Offline Domain Join Server specifically for requests from the MDM Server
443	HTTPS	For sending messages from MDM Server to notification services and identity service.	Outbound from MDM server to:

Port Number	Type	Purpose	Direction
		• Android MDM Server to Google APIs • Apple MDM Server to APNs • Windows MDM Server to WNS [1]	• WNS • Google APIs • APNs • Azure Active Directory • Offline Domain Join server
5671	AMQP	MDM Plugin receives the asynchronous notifications that the MDM Server gets from the enrolled devices through this port. This inbound port to the MDM Server must be opened for the Plugin Portal server to establish the session and subsequently receive the device notifications.	Inbound to the MDM Server from from Plugin Portal server
8443	HTTPS	For sending HTTPS requests to the MDM Server REST API.	Outbound from Plugin Portal server and WebUI to MDM Server
636	LDAPS	For Active Directory to securely authenticate end users during enrollment.	Outbound from MDM Server to the Customer LDAP
389	LDAP	For Active Directory insecure authentication of end users during enrollment. **Note:** In case the Active Directory secure port is not enabled, the default insecure port is 389. For best results, use the LDAPS (secure communication) with Active Directory.	Outbound from the MDM Server to Customer LDAP
2195*	TCP	Backup port for sending messages from the MDM Server to APNs.	Outbound from the MDM Server to the APNs Server (Internet).
2196*	TCP	Used by the MDM Server to connect to APNs for feedback.	Outbound from the MDM Server to the APNs Server (Internet).
5223	TCP	For sending messages to APNS from the computers in your network.	Outbound from Mac devices (whichever network they are on) to the APN Server (Internet).

[1] The WNS push messages are sent to https://wns2-bl2p.notify.windows.com/. *For Windows WNS Firewall recommendations, see https://docs.microsoft.com/en-us/windows/apps/design/shell/tiles-and-notifications/firewall-allowlist-config*

Port Number	Type	Purpose	Direction
8080	TCP	For internal NDES configuration or as configured in the SCEP URL in the fixlet Configure settings for SCEP functionality on MDM server	Outbound from MDM Server to SCEP

*To ensure reliable Apple MDM server communication, allow outbound connections from the MDM Server to the Apple 17.0.0.0/8 block over TCP ports 2195 and 2196.

BigFix PlugIn and MDM SSL certificates and keys

SSL certificates and keys are required to authenticate the BigFix MDM PlugIns to the MDM Server.

These certificates and keys must be generated through the `BESAdmin` command. The generated SSL certificates and keys are stored in the directory that you specify in the BESAdmin command.

Note: You must have a reachable DNS host name to run the commands in the BESAdmin tool to generate certificates.

To generate SSL certificates on a Windows BigFix root server, run this command:

```
BESAdmin.exe /generateplugincertificates /certificatespath:<path-to-store-certs>
 [/commonname:<CN-for-server-and-client-cert>]
```

To generate SSL certificates on a Linux BigFix root server, run this command:

```
BESAdmin.sh -generateplugincertificates -certificatespath=<path-to-store-certs>
 [-commonname:<CN-for-server-and-client-cert>
```

Note:

- For *commonname*, use the FQDN name of the MDM Server.
- These commands work only if *`path-to-store-certs`* directory exists.

The following SSL certificates are generated in the folder that you created. You have to use these SSL certificates and keys when you install the MDM Plugin and MDM Server.

- `ca.cert.pem`
- `client.cert.pem`
- `client.key`
- `server.cert`
- `server.key`

BigFix MDM server TLS certificate and key

The BigFix MDM server requires a CA-signed TLS certificate to protect the communications from the endpoint to the BigFix MDM server. The SSL certificate is deployed through the MDM Server installation in the WebUI.

BigFix MDM server installation requires the following information:

- MDM Server TLS certificate chain with a `.crt` or `.pem` extension
- MDM Server TLS private key with a `.key` extension
- MDM Server TLS private key password

Note: Depending on the trusted CA you use, if this information is in a format other than the required format, you need to work offline to get it in the required format before installing the MDM server.

See additional notes at BigFix MDM Server TLS Certificate Content (on page 40).

Related information

Wildcard certificates *(on page 40)*

Apple Push Notification certificates

The Apple Push Notification Service (APNs) is used to notify Apple devices to check in with their assigned MDM Server. For your MDM Server to communicate with Apple device using the APNs, your MDM Server needs to be configured with an Apple push certificate and key. Obtaining an APNs certificate is only required if you plan to deploy the BigFix MCM Apple service or BigFix PlugIn.

To obtain a push certificate from Apple, as a BigFix Administrator, you require an Apple ID, which must be associated with your enterprise. You can create an Apple ID on the Apple ID web portal. You must use a company email address for this Apple ID, and ideally, it should resolve to a distribution list that is monitored by more than one person. The Apple ID is needed at the step when you login to the Apple portal to create a push certificate for your MDM Server. The push certificate that you obtain is tied to that Apple ID.

Generating an APNs certificate requires the following steps:

1. Create a CSR request
2. Have Bigfix sign the CSR request (via email to BFAppleCSR@hcl.com)
3. Have Apple countersign the CSR and generate the APNs certificate through the Apple portal

For the commands and details for executing the above steps, see Generating APNs certificate (on page 25)

The APNs certificate and keys can then be uploaded to the BigFix MDM server via the WebUI. See Install BigFix MDM Service for Apple

WNS credentials

BigFix Windows MDM service must be authenticated with Windows Notification Service (WNS) credentials. Once authenticated, the Windows MDM service receives a token that it can use to initiate communication with the Windows MDM devices.

To learn more about WNS, see https://docs.microsoft.com/en-us/windows/client-management/mdm/push-notification-windows-mdm

To authenticate BigFix MDM server with Microsoft WNS server, organizations must provide the following information in a JSON format.

```
grant_type = “client_credentials”

client_id = SID<Provided by WNS>

client_secret = <Provided by WNS>

scope= “notify.windows.com”
```

For detailed steps for creating WNS credentials, see Generating WNS credentials (on page 20)

The WNS credentials can then be uploaded to the BigFix MDM server via the WebUI. See Install BigFix MDM Service for Windows.

Google Enterprise Credentials

Google Enterprise Credentials are needed to utilize Android Management functionalities.

If you use non-Google Workspace (formerly non-G-suite) account, to generate Google Enterprise credentials, you must Enroll to Managed Google Play Accounts enterprise. To do that,

1. While Installing BigFix MDM Service for Android, provide the Android Server Admin Credentials.
2. Log into the Admin Configuration page (Example: https:/<MDM_ENROLLMENT_SERVER>/config) with the credentials specified during installation and generate Google Enterprise credentials. For detailed steps, see Enroll to Managed Google Play Accounts enterprise *(on page 30)*.

Installing BigFix PlugIn Portal

The PlugIn Portal provides the server infrastructure for MCM and BigFix Mobile.

MCM and BigFix Mobile requires the PlugIn Portal and one or more MDM PlugIns to be installed in the PlugIn Portal.

Installing the PlugIn Portal

To install the PlugIn Portal as a service on a BigFix client, For instructions on PlugIn Portal installation and base configuration, see Installing the PlugIn Portal. This installs the BigFix PlugIn Portal on the selected targets.

The PlugIn Portal is typically installed in the following directories:

- Windows:
  - C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal
- Linux:
  - /var/opt/BESPluginPortal
  - /opt/BESPluginPortal

Note:

- You might have several PlugIn Portal in your environment; but on a specific target computer, you can have only one.
- You can have only one Windows MDM plugin, one Apple MDM plugin, and one Android MDM plugin in an MDM deployment. Therefore, if you are using two PlugIn Portal, for example, each PlugIn Portal can contain only one MDM Plugin (Windows or Apple).

The PlugIn Portal needs an MDM PlugIn to work for MCM. The MDM PlugIn installation becomes relevant only if there is a valid PlugIn Portal present on the system. The local BigFix client evaluates the Tasks periodically to check if they become relevant.

Installing MDM services

You can set up MDM services for Windows, Apple, and/or Android through WebUI.

- Install Docker Engine, Docker Compose, and OpenSSL on the intended MDM server.
- Ensure that you have one of these BigFix user roles:
  - A Master Operator (MO) with visibility over the MDM Server target machine and visibility of the BESUEM site
  - An Administrator with the privilege to run the installation.
- Install the BigFix client on the target computer in which you want to install the MDM server.

Installing BigFix MDM Service for Apple, Windows, and/or Android completes these activities:

1. Downloads a set of docker images from software.bigfix.com which is required for the MDM installation.
2. Installs the services and certificates including the PlugIn certificates *(on page 11)* and the TLS certificate on which the server runs, and the Apple Push certificate if you are installing BigFix MDM Service for Apple.
3. Applies all required configurations.

For instructions on setting up the MDM server through WebUI, see the following links:

- Install BigFix MDM Service for Windows
- Install BigFix MDM Service for Apple
- Install BigFix MDM Service for Android

Installing MDM Plugin

You need MDM Plugin to set up a connection between the MDM Servers and the BigFix Plugin Portal. MDM Plugin communicate with the MDM Server through REST APIs and the AMQP protocol using client certificates. For instructions on setting up through WebUI, see WebUI User's Guide.

Ensure that the server host is running the Plugin Portal and that the BigFix agent is running locally. For details about installing the BigFix Client, see Installing the BigFix components.

Three versions of the MDM Plugin are available: Apple, Windows, and Android. For detailed instructions on how to install the respective MDM Plugin, see:

- Install MDM Plugin for Windows
- Install MDM Plugin for Apple
- Install MDM Plugin for Android

Note: These installation procedure require credentials, specifically from CA cert, the client cert, and the client key that is generated from BESAdmin.sh. For details, see BigFix PlugIn and MDM SSL certificates and keys *(on page 11)*

Chapter 4. On-premises Upgrade

If you already have BigFix MCM and/or BigFix Mobile 2.x deployment, and if you want to upgrade to version 3.x, you can find the instructions here.

Before you begin:

- You must be a Master Operator to perform this task through WebUI.
- You need Plugin Portal version 10.0.2 or later to update the MDM Plugins to the latest version. For all the MCM 3.1 features to work, you need Plugin Portal version 10.0.8 or later.

Step 1: Update MDM Server

To update MDM Server:

1. From the WebUI main page, click **Apps > MCM**.
2. On the Modern Client Management page, click **Admin**.
3. On the Admin page, from the left navigation, under MDM Server, click **Update**
4. In the Target Devices section, click **Edit Devices**. A list of the available servers that need an update is displayed. Select the required servers and click **OK**.
5. Review the number of servers selected and click **Deploy**. WebUI runs the update on the targeted servers.

Step 2: Update MDM Plugins

To update MDM Plugins:

1. From the WebUI main page, click **Apps > MCM**.
2. On the Modern Client Management page, click **Admin**.
3. On the Admin page, from the left navigation, under MDM Plugins, click **Update**
4. In the Target Devices section, click **Edit Devices**. A list of the available devices that need an update is displayed. Select the required devices and click **OK**.
5. Review the number of servers selected and click **Deploy**. WebUI runs the update on the targeted servers.

Step 3: Add Credentials

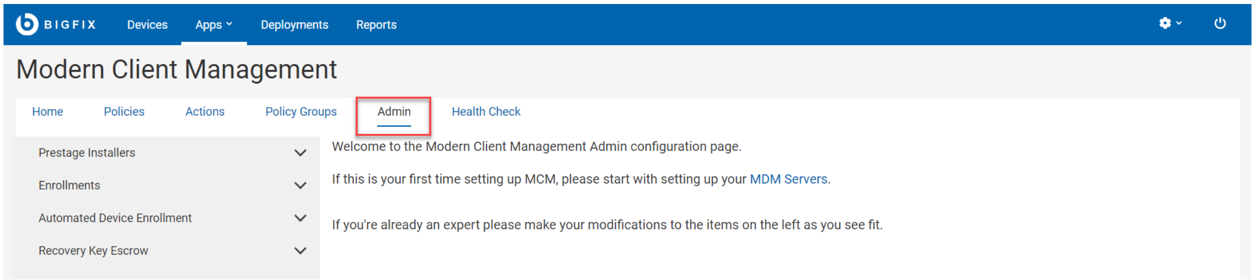
If you are upgrading MCM from 2.x to 3.x, to establish direct connectivity between WebUI and the MDM Server, you must upload the same set of server credentials and client credentials that were originally uploaded while installing MDM server and MDM Plugin respectively. For instructions to add credentials, see Add Credentials.

Note: If you do not add appropriate credentials after upgrading, no connectivity between WebUI and MDM server for any of the Smart Group or Manage Capability functions.

Chapter 5. Configuring BigFix MCM and BigFix Mobile

After the MCM components are set up, there are additional configuration options available to enable features like Bulk Enrollment for Windows, DEP policies for macOS, or prestage installers for Windows and MacOS MDM endpoints.

To configure MCM, from the WebUI main page, click **Apps > MCM** and from the Modern Client Management page, select **Admin**.



Welcome to the Modern Client Management Admin configuration page.

If this is your first time setting up MCM, please start with setting up your MDM Servers.

If you're already an expert please make your modifications to the items on the left as you see fit.

Depending on the operating system and enrollment type, you can navigate to the configuration option to complete these configuration tasks:

- Prestage macOS BigFix installer
- Prestage Windows BigFix Installer
- Prestage an Application
- Set up Apple App Store (iOS and iPadOS) and Google Play Store (Android) Associations
- Create Windows Provisioning Package
- Designate Provisioning Package Generation Point
- Configure Windows Autopilot Terms of Service
- Generate Encryption Recovery Key Escrow Certificate
- Setup Recovery Key Escrow Plugin
- Manage Automated Device Enrollment Policies

Appendix A. Support

For more information about this product, see the following resources:

- BigFix Support Portal
- BigFix Developer
- BigFix Playlist on YouTube
- BigFix Tech Advisors channel on YouTube
- BigFix Forum

Appendix B. Reference Information

This section contains more detailed information about some of the essential installation aspects.

Generating WNS credentials

This document describes how to get Windows Notification Service (WNS) credentials that you can upload during installing or upgrading Windows MDM server.

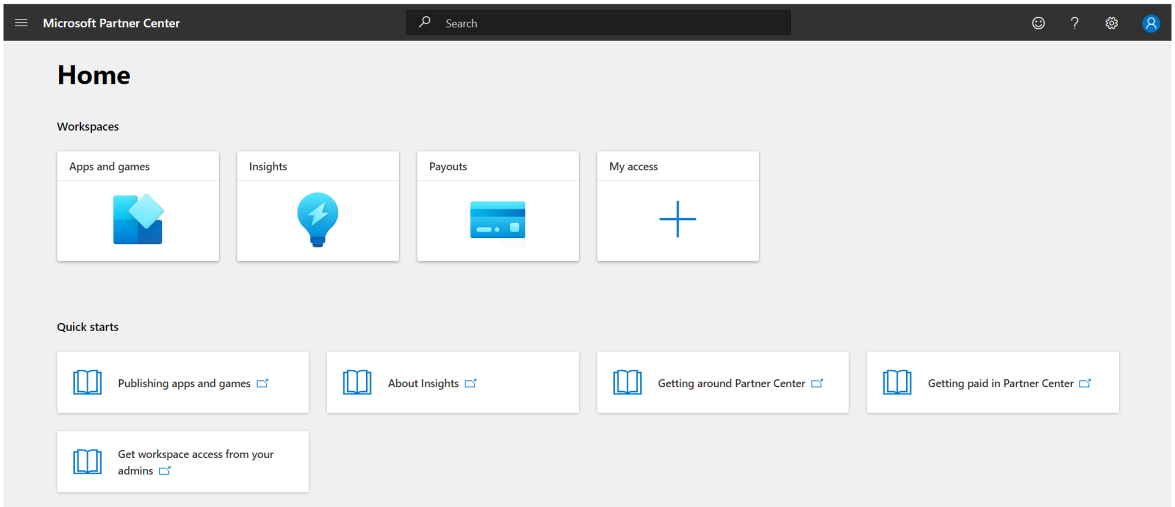
The organization must have a paid Microsoft developer account to create WNS credentials.

Note: For detailed instructions about how to create a paid Microsoft developer account, see https://bigfix-wiki.hcltechsw.com/wikis/home?lang=en-us#!/wiki/BigFix%20Wiki/page/How%20to%20create%20a%20Microsoft%20developer%20account

To establish the communication with the enrolled devices, the windows MDM server must know the credentials of WNS server through which windows MDM server will communicate with enrolled devices to apply polices and actions, seehttps://docs.microsoft.com/en-us/windows/client-management/mdm/push-notification-windows-mdm.

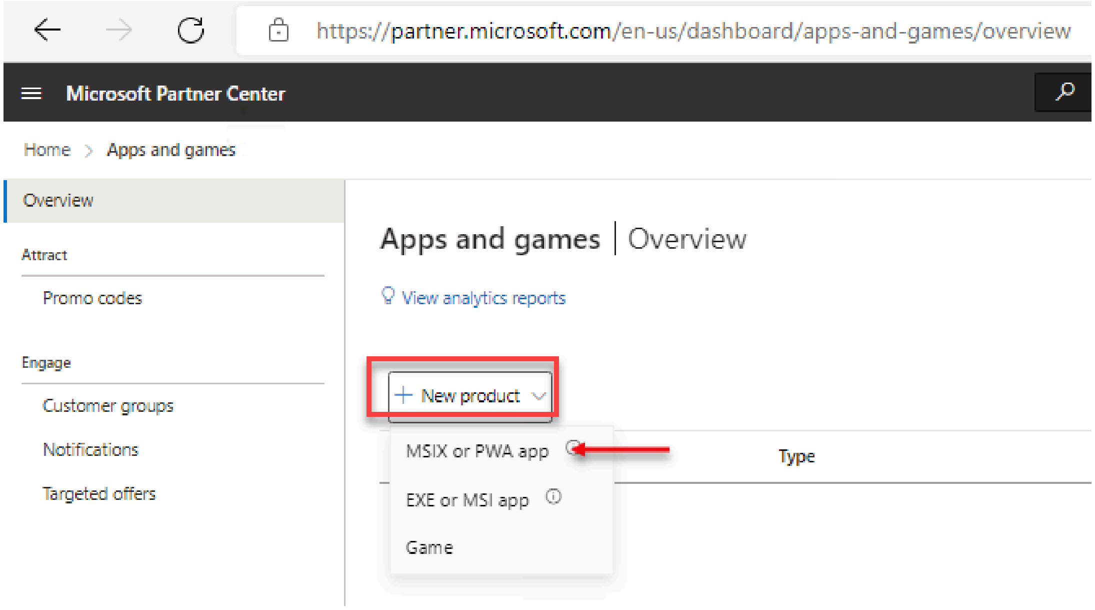
To get WNS credentials for the MDM server, complete the following steps:

1. Login to Microsoft Partner Center: Open the URL *https://partner.microsoft.com/en-us/dashboard* and enter the Microsoft developer account credentials. The following page is displayed.

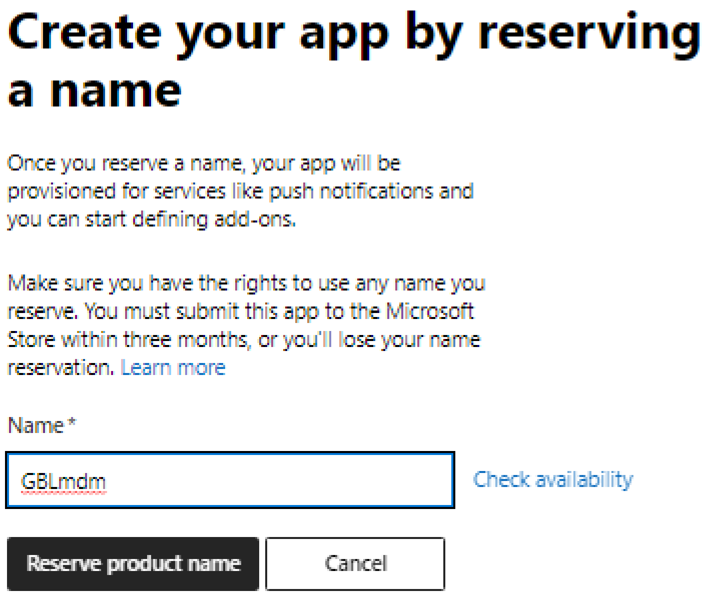

Microsoft Partner Center
Search
Home
Workspaces
Apps and games
Insights
Payouts
My access
Quick starts
Publishing apps and games
About Insights
Getting around Partner Center
Getting paid in Partner Center
Get workspace access from your admins

2. Click **Apps and games**. The Apps and games Overview page is displayed.
3. Create an app and get WNS push credentials (client secret, PFN, and SID). To do that, complete the following steps.

a. From the Overview page, click **New product** and select **MSIX or PWA app**.



b. On the **Create your app by reserving a name** page, enter an appropriate name for the application (which will be the WNS server name) and click **Reserve product name**.



Create your app by reserving a name

Once you reserve a name, your app will be provisioned for services like push notifications and you can start defining add-ons.

Make sure you have the rights to use any name you reserve. You must submit this app to the Microsoft Store within three months, or you'll lose your name reservation. Learn more

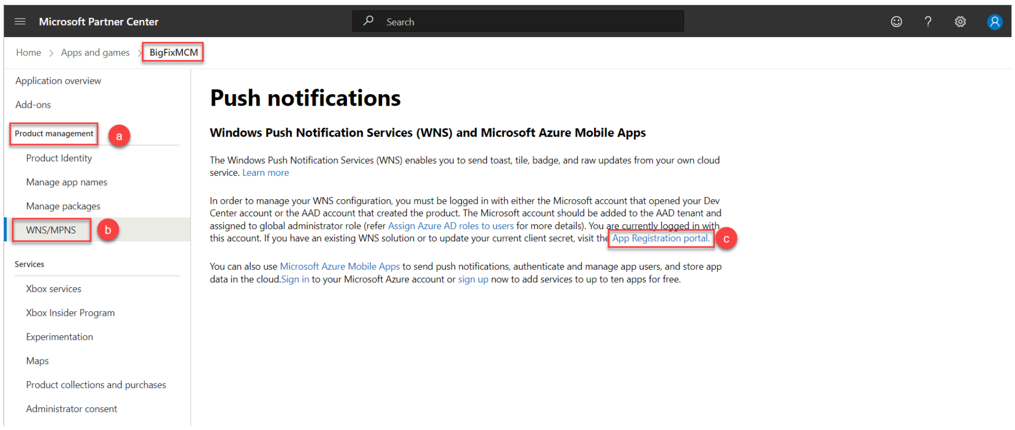
Name *

GBLmdm

4. Create `wnscredentials.json` file.

a. Go to **Home > Apps and Games**, and from the product list, select the application you have created.

i. From the product page, navigate to **Product Management > WNS/PNS** and click the **App Registration portal** link.



Push notifications

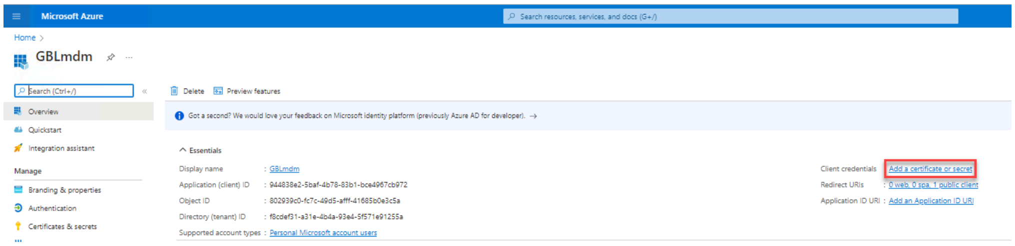
Windows Push Notification Services (WNS) and Microsoft Azure Mobile Apps

The Windows Push Notification Services (WNS) enables you to send toast, tile, badge, and raw updates from your own cloud service. Learn more

In order to manage your WNS configuration, you must be logged in with either the Microsoft account that opened your Dev Center account or the AAD account that created the product. The Microsoft account should be added to the AAD tenant and assigned to global administrator role (refer Assign Azure AD roles to users for more details). You are currently logged in with this account. If you have an existing WNS solution or to update your current client secret, visit the App Registration portal.

You can also use Microsoft Azure Mobile Apps to send push notifications, authenticate and manage app users, and store app data in the cloud.Sign in to your Microsoft Azure account or sign up now to add services to up to ten apps for free.

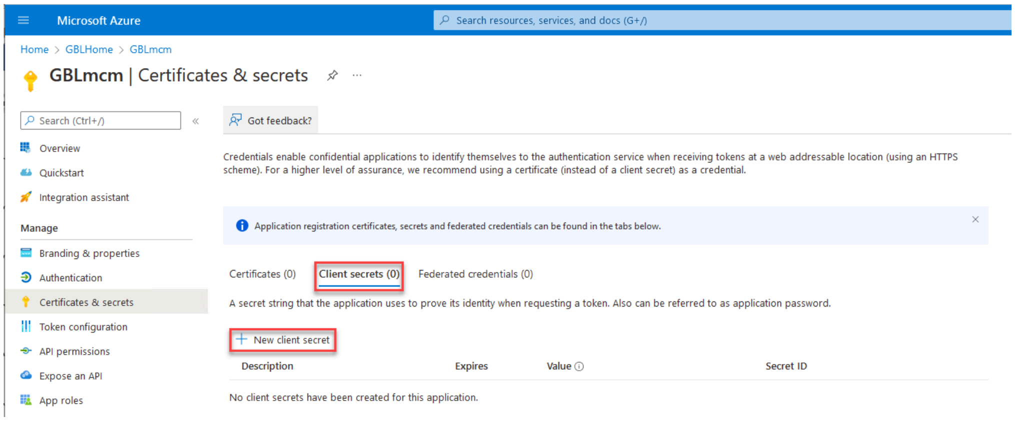
ii. Microsoft Azure portal page for your app is displayed. Click the **Client credentials** link to add a certificate or secret.



Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Display name : GBLmdm
Application (client) ID : 944838e2-5baf-4b78-83b1-bce4967cb972
Object ID : 802939c0-fc7c-49d5-afff-41685b0e3c5a
Directory (tenant) ID : f8cdef31-a31e-4b4a-93e4-5f571e91255a
Supported account types : Personal Microsoft account users
Client credentials : Add a certificate or secret
Redirect URIs : 0 web, 0 spa, 1 public client
Application ID URI : Add an Application ID URI

iii. On the Certificates & secretes page, under **Client secrets** tab, click **+ New client secret**.



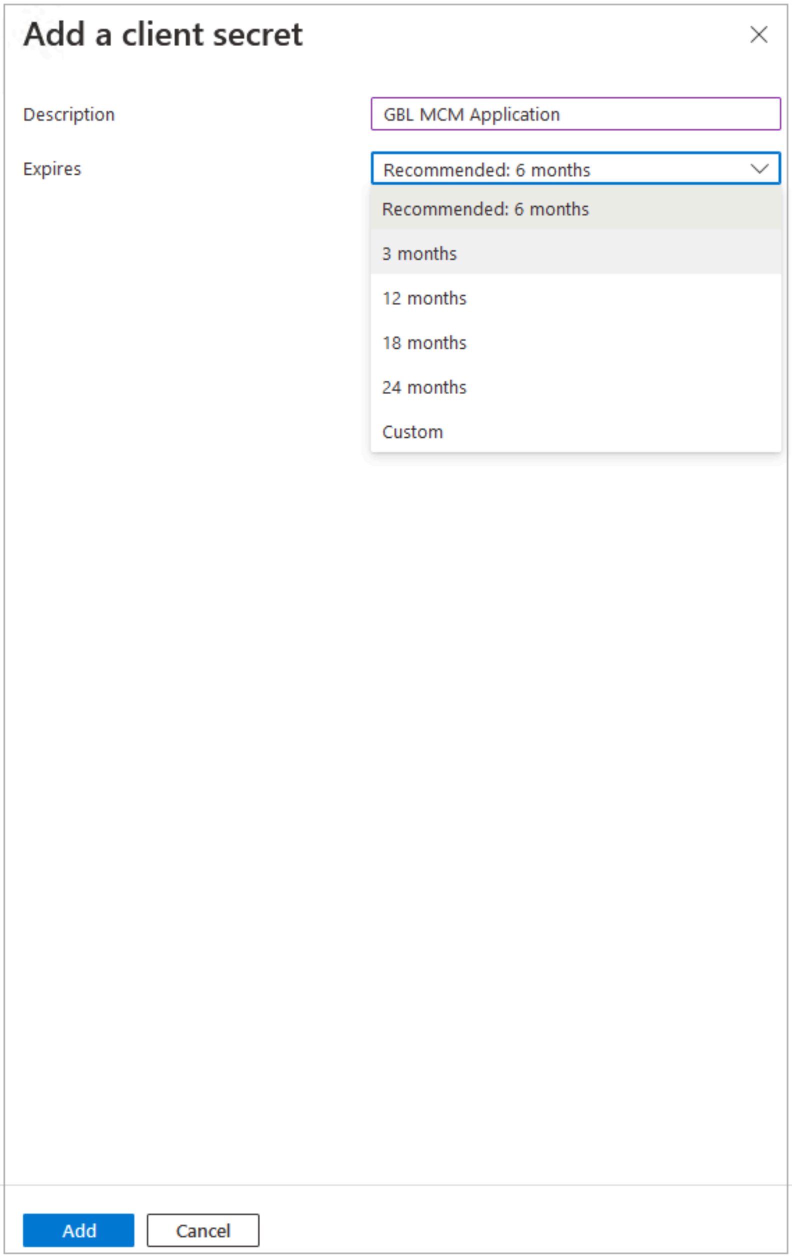
Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

No client secrets have been created for this application.

iv. Enter **Description** of the client secret, click the **Expires**drop-down and select the validity period of the client secret, and click **Add**.

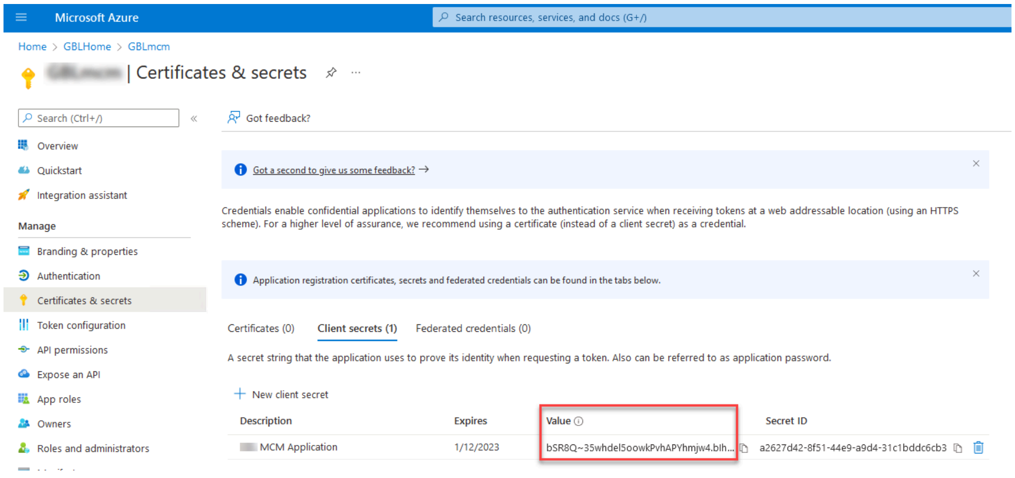


Add a client secret

Description: GBL MCM Application

Expires: Recommended: 6 months

v. Copy the **Value** of the Secret ID to use it as the `client_secret` value in the `WNScredential.json` file.



[redacted] | Certificates & secrets

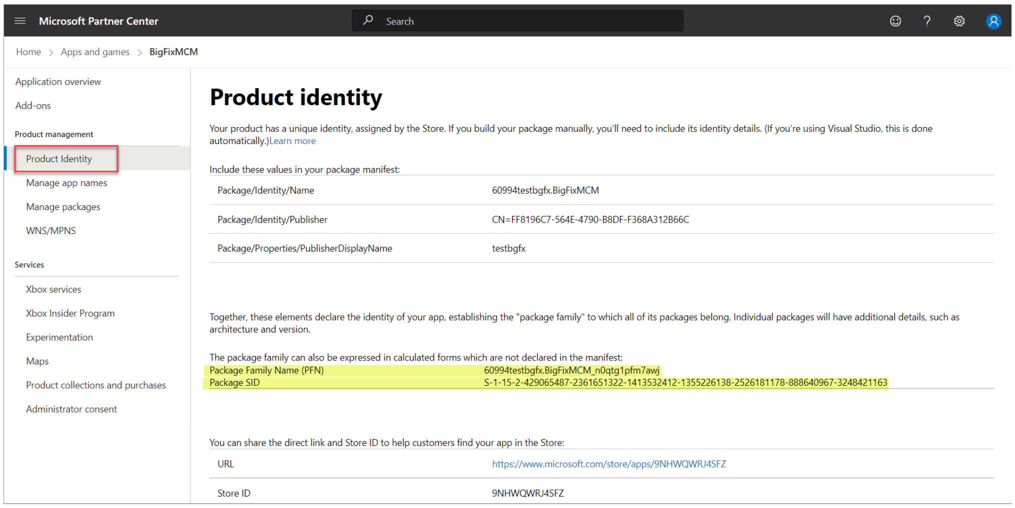
Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value	Secret ID
MCM Application	1/12/2023	bSR8Q~35whdel5oowkPvhAPYhmjw4.blh...	a2627d42-8f51-44e9-a9d4-31c1bddc6cb3

vi. Navigate to **Product Management > Product Identity** and copy the PFN and Package SID to add them to the `WNScredential.json` file.



Product identity

Your product has a unique identity, assigned by the Store. If you build your package manually, you'll need to include its identity details. (If you're using Visual Studio, this is done automatically.)Learn more

Include these values in your package manifest:

Package/Identity/Name	60994testbgfx.BigFixMCM
Package/Identity/Publisher	CN=FF8196C7-564E-4790-B8DF-F368A312B66C
Package/Properties/PublisherDisplayName	testbgfx

Together, these elements declare the identity of your app, establishing the "package family" to which all of its packages belong. Individual packages will have additional details, such as architecture and version.

The package family can also be expressed in calculated forms which are not declared in the manifest:

Package Family Name (PFN)	60994testbgfx.BigFixMCM_n0qtg1pfm7awj
Package SID	S-1-15-2-429065487-2361651322-1413532412-1355226138-2526181178-888640967-3248421163

You can share the direct link and Store ID to help customers find your app in the Store:

URL	https://www.microsoft.com/store/apps/9NHWQWRJ4SFZ
Store ID	9NHWQWRJ4SFZ

vii. Furnish the copied information in the following format and save the file as `wnscredentials.json`.

```
{
"client_id": "ms-app://<Package SID>",
"client_secret":"<Application Secrets>",
"PFN":"<PFN>"
}
```

The `wnscredentials.json` file is created that can be uploaded while Installing BigFix MDM Service for Windows to establish the communication between the MDM server and Windows devices.

Generating APNs certificate

To obtain a push certificate from Apple, complete these steps:

1. **Create CSR Request**: In the command-line interface on a Linux server, run the following command to create a CSR for the push certificate using the openssl tool:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout <PUSHCERTNAME>_temp.key -out <PUSHCERTNAME>.csr -subj
    "/C=US/CN=<HOSTNAME>/emailAddress=<EMAILADDRESS>"
   ```

   Note:

   - Replace `<PUSHCERTNAME>` with a name of your choice.
   - `<EMAILADDRESS>` must be unique to your organization and is for reference purposes only. This email address forms part of the certificate subject line and could be used in future by Apple to contact whoever will be considered the administrative contact for the push certificate. It is recommended to use this email address of the Apple ID in the subsequent certificate creation step.
   - `<HOSTNAME>` must be the FQDN of the server on which the MDM server runs. This records the FQDN of the server, which uses the Push Certificate in the subject line of the certificate.

2. **Encrypt APNs private key**: Run the following command to encrypt the private key:

   ```
   openssl rsa -des3 -in <PUSHCERTNAME>_temp.key -out <PUSHCERTNAME>.key
   ```

   Enter the encrypted private key pass phrase of your choice when prompted. You will then be asked to verify it.

   Important:

   - Save the generated files `<PUSHCERTNAME>.csr` and `<PUSHCERTNAME>.key` along with the private key pass phrase in at a safe location. You will need to use these for subsequent push certificate renewals. The CSR and private key pass phrase used at the time of initial certificate creation will be needed to complete the renewal process, so it is very important these are retained in a safe place.
   - Apple push certificates have a one-year lifetime. The WebUI Modern Client Management dashboard notifies the WebUI user if certificates are nearing expiry. You need to Renew APNs certificate and update Apple MDM service *(on page 26)* annually when it gets close to expiry, and not create a brand new one, otherwise any enrolled devices will be orphaned.

3. **Request CSR signatures**: Send the `CSR` file to BFAppleCSR@hcl.com.

   Important: Include your HCL Customer ID or BigFix server serial number in the body of the email. This is necessary to authorize the request and validate entitlement to MCM or BigFix Mobile.

An HCL-signed version of the CSR file, plus additional instructions from BFAppleCSR@hcl.com will be returned to the sender's email address within one business day. Follow the instructions in that email to obtain the required file through your Apple Developer account.

4. **Generate the Push Certificate**
    a. Log in to the Apple Push Certificates Portal using your Apple ID and click **Create a Certificate**.
    b. Upload the HCL-signed version of the CSR file obtain a provider certificate from Apple.
    c. Download the push certificate (.pem).
    d. Save the push certificate at a safe location.

        You will need to supply this push certificate, and the associated private key and passphrase when you install the Apple MDM Server.

Renew APNs certificate and update Apple MDM service

You can renew your APNs certificate within the validity period before expiration.

Apple push certificates have a one-year lifetime. The WebUI Modern Client Management dashboard notifies the WebUI user when certificates are within 30 days of expiry.

Important: If the APNs certificate has already expired, you must set up a new certificate. See, Generating APNs certificate *(on page 25)*. If you generate a brand new certificate, already enrolled devices will be orphaned. To avoid this, you need to renew these APNs certificate annually when it gets close to expiry.

To renew the APNs and update the certs in the Apple MDM service, complete the following:

1. **Request CSR signatures**: Send the CSR generated initially *(on page 25)*to BFAppleCSR@hcl.com.

    Important: Include your HCL Customer ID or BigFix server serial number in the body of the email. This is necessary to authorize the request and validate entitlement to MCM or BigFix Mobile.

    An HCL-signed version of the CSR file, plus additional instructions from BFAppleCSR@hcl.com will be returned to the sender's email address within one business day. Follow the instructions in that email to obtain the required file through your Apple Developer account.
2. **Renew the Push Certificate**
    a. Log in to the Apple Push Certificates Portal using the same Apple ID with which you generated the APNs initially.
    b. Locate the certificate you want to update and click **Renew**.
    c. Upload the HCL-signed version of the CSR file obtain a provider certificate from Apple.
    d. Download the push certificate (.pem).
    e. Save the push certificate at a safe location.
3. **Supply this push certificate into Fixlet 409** Update Apple push certificate *(on page 27)* to update the certs in the Apple MDM service.

Update Apple push certificate

Use Fixlet 409 `Update Apple Push Certificate` to update the Apple Push certificate on the BigFix Apple MDM service.

To update complete the following steps.

1. From the BESUEM site, open Fixlet 409 `Update Apple Push Certificate`.
2. In the **Apple Push Certificate PEM content** text box, enter the content of the renewed push certificate (`.pem`) content.
3. Click on the link **here** to update the APNs certificate on the Apple MDM service.

Update Apple Enrollment Certificate before expiration

To continue to manage the enrolled Apple devices without interruption, you must set up the Fixlet "Update Apple Enrollment Profile before Expiration" as a policy action.

Apple Enrollment Certificate

An Apple Enrollment Certificate (Device Identity Certificate) authorizes an MDM device to talk to the MDM Server. All requests from the MDM devices are signed with this Device Identity Certificate. At the time of MDM enrollment, when Apple device communicates with the MDM Server, the MDM Server generates and assigns Unique Device Identity Certificates (or SCEP certificates) to each device. The MDM Server ensures that requests coming from each device are signed by the correct Device Identity certificate; if not, the requests are ignored.

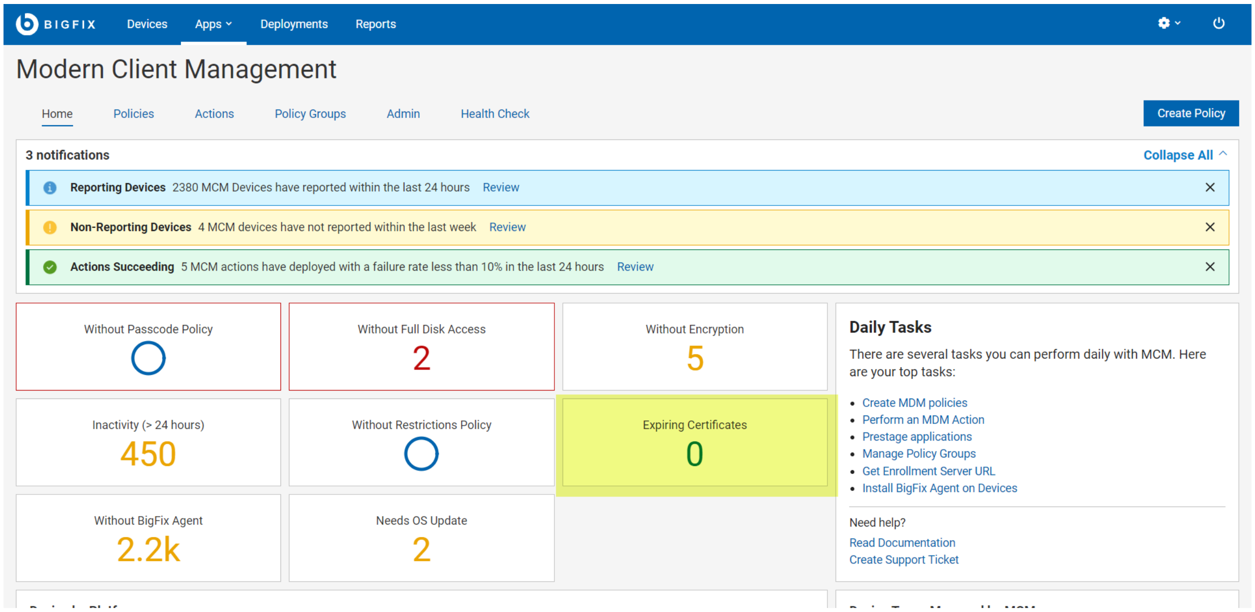
This certificate has one year validity. You must renew the existing Apple Device Identity Certificate before the expiration date.

How to identify the devices with expiring Apple Identity Certificates

If you set up the Fixlet `Update Apple Enrollment Profile before Expiration` as a policy action with the intended targets selected, you can get the visibility of the expiration date of Device Identity Certificates for the targeted Apple devices. The WebUI Modern Client Management dashboard notifies the WebUI user about the devices with certificates nearing expiry.

CAUTION: If this Fixlet is not set up, you cannot track the expiry dates through WebUI dashboard.

The following image shows the WebUI dashboard with “Expiring Certificates” tile that shows the number of devices with expiring certificates. Clicking on the number shows the list of devices within 45 days of expiry.



Reporting Devices 2380 MCM Devices have reported within the last 24 hours Review

Non-Reporting Devices 4 MCM devices have not reported within the last week Review

Actions Succeeding 5 MCM actions have deployed with a failure rate less than 10% in the last 24 hours Review

Without Passcode Policy ○	Without Full Disk Access 2	Without Encryption 5
Inactivity (> 24 hours) 450	Without Restrictions Policy ○	Expiring Certificates 0
Without BigFix Agent 2.2k	Needs OS Update 2	

Daily Tasks

There are several tasks you can perform daily with MCM. Here are your top tasks:

- Create MDM policies
- Perform an MDM Action
- Prestage applications
- Manage Policy Groups
- Get Enrollment Server URL
- Install BigFix Agent on Devices

Need help?
Read Documentation
Create Support Ticket

The Update Apple Enrollment Certificate before expiration Fixlet

This Fixlet is available under BESUEM site. If you have Apple Devices in your MCM deployment, you must set up this Fixlet as a policy action with the correct targets selected.

When set up as a policy action, this Fixlet does the following actions:

- It looks for all devices where the Device Identity Certificate is within 45 days of expiry, which means it has been almost a year since the device last received an enrollment profile and any updated certificates.
- Displays devices with less than 45 days of expiration of their device identity certificates on the main MCM Dashboard in a tile showing Expiring Certificates.
- Initiates an update enrollment profile action to the relevant Apple devices. If the devices are up and running and check in at least once a day, the policy action deploys the latest enrollment profile (with the latest certificates) onto those devices. This auto-renews the certificates for the devices nearing expiry date and ensures these devices do not stay under the tile “Expiring Certificate” for more than a day. Internally it does the following actions:
    - Provides a new Device Identity Certificate to the device which allows it to operate successfully for another year
    - Pushes the latest TLS certificate’s Intermediate certificate to the device to ensure it is trusted
    - Signs the enrollment profile with the latest available signing certificate to ensure the profiles remain "Verified"

How to set up the Fixlet to auto-renew the certificates

Complete the following steps to set up this Fixlet as a policy action.

1. In the BESUEM site, find the Fixlet `Update Apple Enrollment Certificate before expiration`.
2. Select **Take Action**.
3. Change the **preset type** to `Policy`.
4. Select **Dynamically target by property**.
5. Under the **Execution** tab:
    - Select **Reapply this action**.
    - Select **while relevant**.
    - Select **1 day**

    The policy is set to reapply the action once a day while relevant. There is no limit for reapplications.

! **Important:** The devices identified under “Expiring Certificate” tile must be online and not screen-locked to process the renewal requests before expiration. If the target device does not become active and check in, the certificates can still expire and then the devices can become unmanageable.

Related reference

Apple profile displayed as unverified

Update Settings for Windows MDM Server

Learn how to update BigFix MDM Service for Windows for MCM 2.0 or later.

- You can update organization name, user facing hostname, and LDAP parameter through this Fixlet.
- If you are updating the BigFix MDM Service for Windows from MCM 1.1 to MCM 2.0, you must add WNS credentials via BESUEM Fixlet 407 `Update Settings for Windows MDM Server`. Only after that, the Fixlet to update to MCM 2.0 becomes relevant. You can then update the MDM server through Fixlet or through WebUI.

To add the WNS credentials, run Fixlet 407 `Update Settings for Windows MDM Server` with the following information:

1. Enter the organization name. While enrolling a device, the organization name is displayed to the users along with the rest of the profile information.
2. Enter user facing hostname. This is the hostname of the server that the enrolling devices should be pointing to. The value must be the hostname from a valid URL. For example, enter `mdmserver.deploy.bigfix.com`.
3. Enter WNS Credentials JSON: Copy and paste the content of the `wnscredentials.json` file as created in Generating WNS credentials *(on page 20)*
4. Enter the following LDAP parameters you want to change.
5. Click on the link “here” to run the Fixlet.

Bootstrap tokens for Apple devices

In macOS 10.5 and above, bootstrap tokens are used for granting secure tokens to user accounts and performing certain operations. For example, on a Mac computer with Apple silicon, the bootstrap token, if available, can be used to authorize the installation of both kernel extensions and software updates when managed using MDM.

See https://support.apple.com/en-ca/guide/deployment-reference-macos/apdff2cf769b/web for more details.

If organizations have a specific use case that requires bootstrap token support, MCM provides it by notifying the device that MCM supports this feature at the time of enrollment and by caching and then retrieving these bootstrap tokens on demand for any further operations on the device that need bootstrap tokens.

Update Enrollment Profile Parameters for Apple MDM Server

You can update MDM enrollment profile parameters for Apple devices that was initially set up while installing the BigFix Apple MDM Server.

After installing the Apple MDM Server, if you want to update the enrollment profile parameters for Apple devices at any point, in BESUEM Fixlet 404 `Update Enrollment Profile Parameters for Apple MDM Server`, provide the updated information for the following parameters and deploy the Fixlet to the targeted systems:

1. Organization Name.
2. User-facing hostname.
3. Access rights. This determines the capabilities of the MDM Server on an enrolled device. The value must be a 4-digit decimal number. For more details on how to calculate the number, see https://developer.apple.com/documentation/devicemanagement/mdm
4. User agreement text.

Enroll to Managed Google Play Accounts enterprise

Learn how to enroll to Managed Google Play Accounts enterprise to manage applications on Android devices.

The work flow to enroll to managed Google Play Accounts enterprise is as follows:

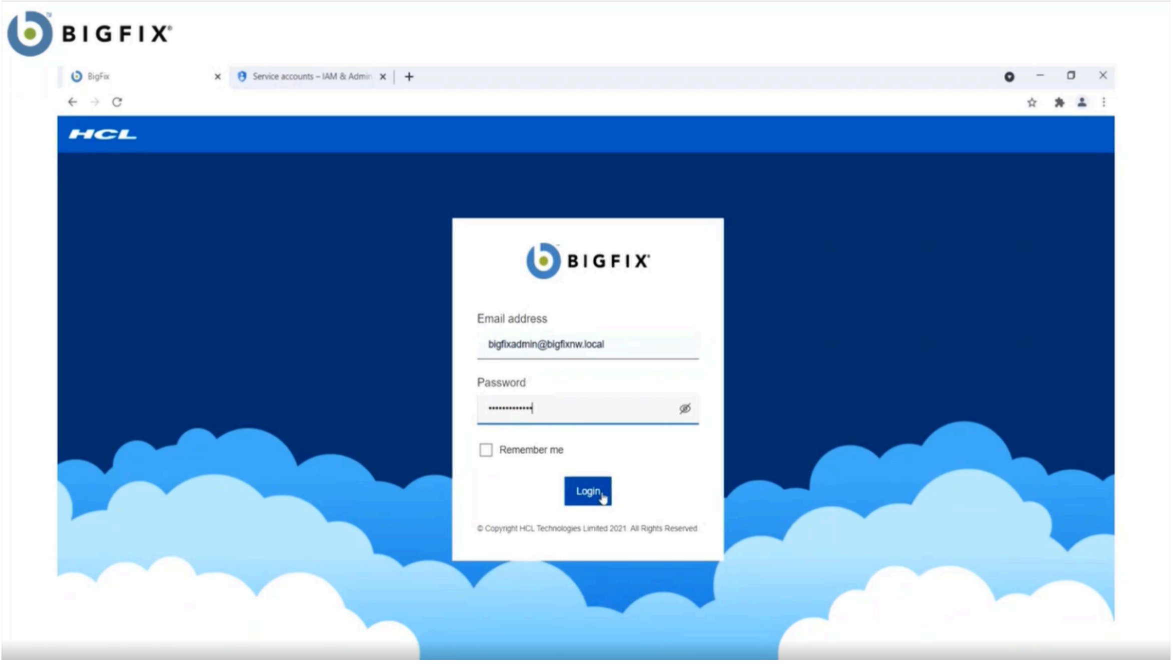
A. Request for enterprise token and upload it: You need an enterprise token (the Google credentials for the master service account), to create service account credentials, and to enable required privileges which is required for enterprise-specific service account. This token has user-restricted privileges and is not a part of MDM server installation. You need to request a token from BigFix MCM Admin Configuration page and upload it to further proceed to create an enterprise service account.

B. Register for an enterprise service account and complete the process to manage Android devices.

A. Request for enterprise token and upload it

As an IT admin, to send request for enterprise token, perform the following steps:

1. Go to the BigFix MCM Admin Configuration (for example, *https://MDM-demo/config*). This URL represents where you installed MDM and is separate from the WebUI. Enter the Android Server Admin user name and password that you have set for non-G-suite account during Android MDM server installation and click **Login**.



Email address

bigfixadmin@bigfixnw.local

Password

••••••••••••

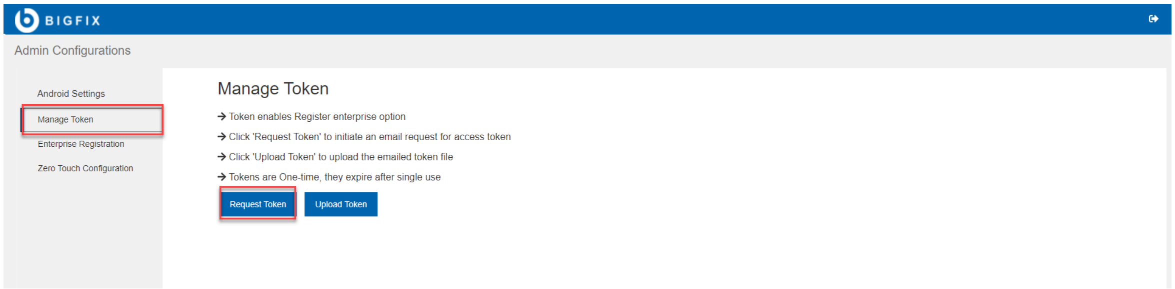
□ Remember me

© Copyright HCL Technologies Limited 2021. All Rights Reserved

Note: If you enter the wrong username/password more than five times consecutively, the site is locked and you cannot log on to BigFix MCM Admin Configuration page. The site gets unlocked after 15 minutes by default. The number of times before the site gets locked and the number of minutes after which the site gets unlocked are configurable through MCM environment variables.

Note: You can change the look and feel of Enrollment UI and Android Server Configuration UI by changing the color, image, logo, brand name and so on. For details, see Rebranding user interfaces *(on page 35)*.

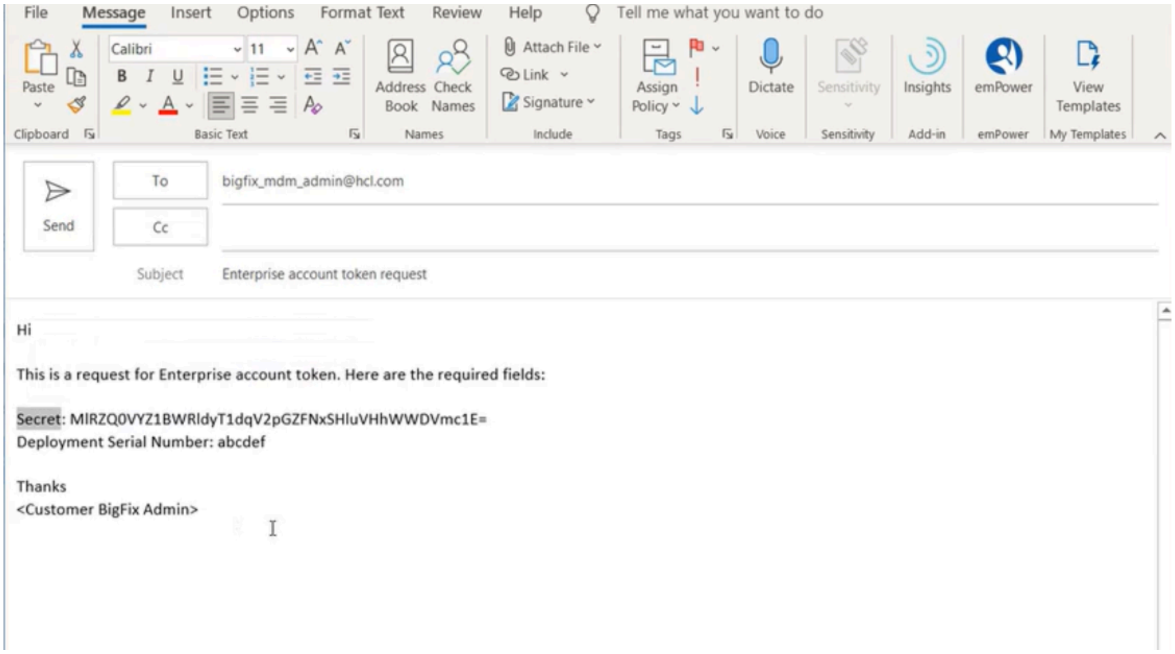
2. From the BigFix **Admin Configurations** page, navigate to **Manage Token** and click **Request Token**.



Manage Token

→ Token enables Register enterprise option
→ Click 'Request Token' to initiate an email request for access token
→ Click 'Upload Token' to upload the emailed token file
→ Tokens are One-time, they expire after single use

This triggers a mail to the logged in BigFix MCM Admin with the generated encoded secret and the deployment serial number.



To: bigfix_mdm_admin@hcl.com

Subject: Enterprise account token request

Hi

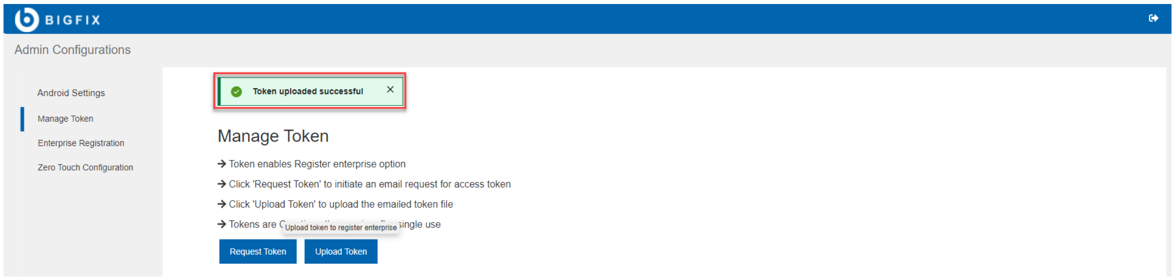
This is a request for Enterprise account token. Here are the required fields:

Secret: MIRZQ0VYZ1BWRldyT1dqV2pGZFNxSHluVHhWWDVmc1E=
Deployment Serial Number: abcdef

Thanks
<Customer BigFix Admin>

3. Send the email with the encoded secret and the deployment serial number to the HCL BigFix MCM Admin (bigfix_mdm_admin@hcl.com). HCL BigFix MCM Admin uses the information, generates the encrypted token file (`FILE_2.enc)`, and mails it back to you.

4. After you receive the encrypted token file from the HCL BigFix MCM Admin, upload it. To do that, in the BigFix MCM Admin Configuration page, navigate to **Manage Token** and click **Upload Token**. On successfully uploading the token, the success message is displayed and under **Enterprise Registration**, the **Register** button is enabled and you can register the enterprise.



Token uploaded successful

Manage Token

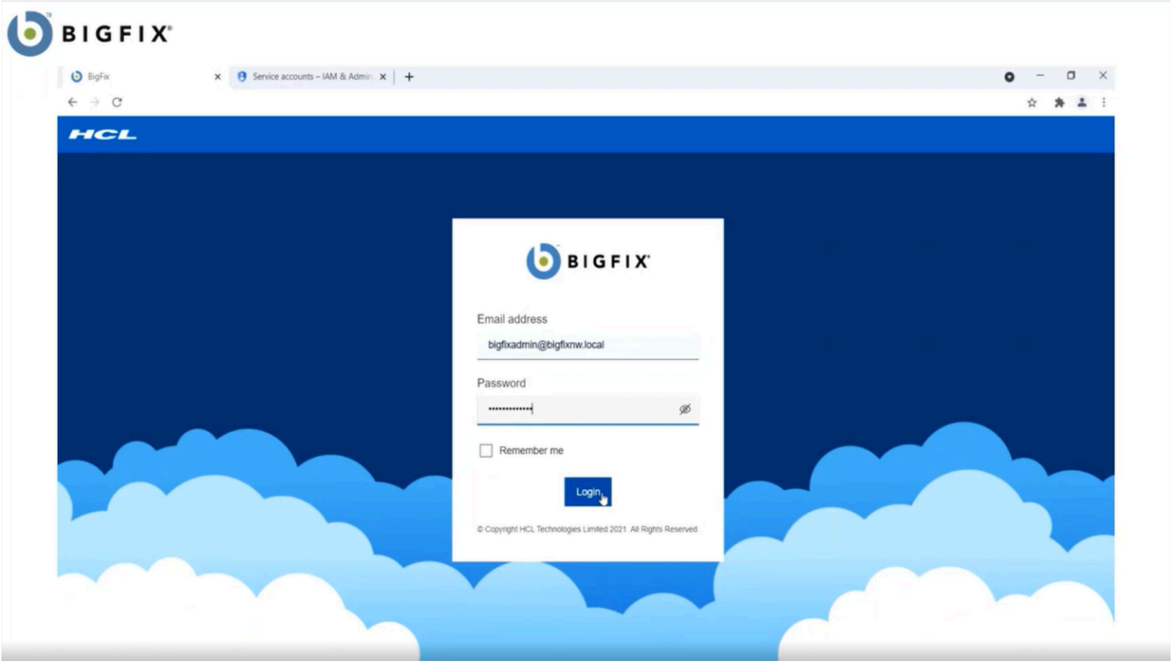
→ Token enables Register enterprise option
→ Click 'Request Token' to initiate an email request for access token
→ Click 'Upload Token' to upload the emailed token file
→ Tokens are C[illegible] single use

Note: The enterprise token can be used just once. After an enterprise service account is created using the token, it gets destroyed.

B. Register for an enterprise service account

As an IT admin, to register your enterprise using the Android Management API, perform the following steps:

1. Go to the BigFix MCM Admin Configuration page (for example, *https://MDM-demo/config*. Note this is separate from WebUI). Enter your organization email ID and password and click **Login**.



Email address

bigfixadmin@bigfixnw.local

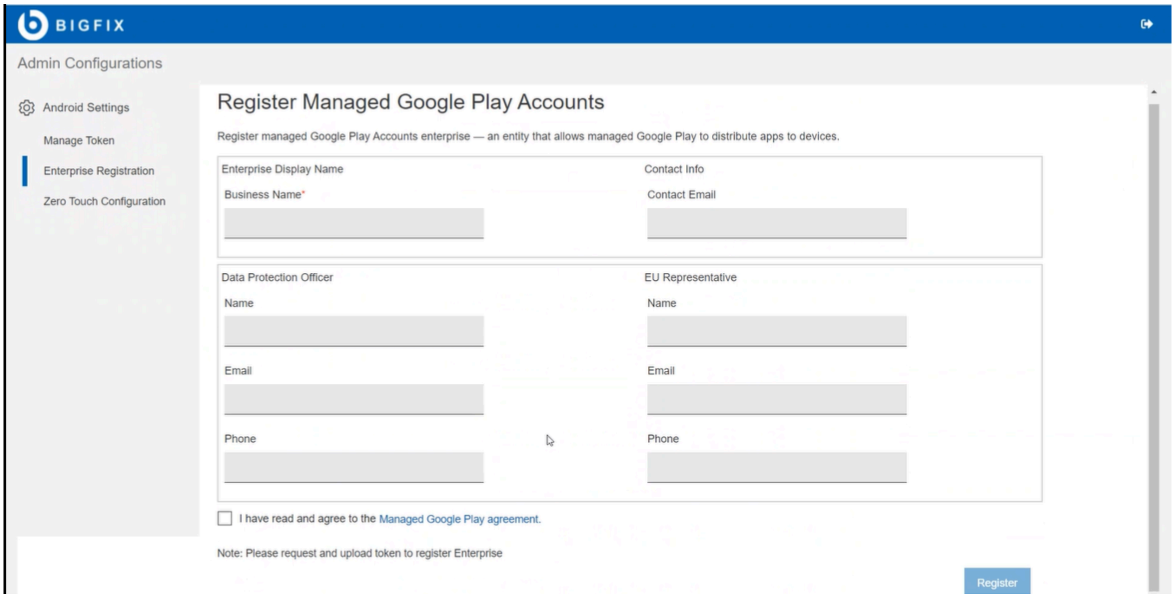
Password

□ Remember me

Login

© Copyright HCL Technologies Limited 2021. All Rights Reserved

2. The following screen appears, Enter the required details.



Admin Configurations

- Android Settings
- Manage Token
- Enterprise Registration
- Zero Touch Configuration

Register Managed Google Play Accounts

Register managed Google Play Accounts enterprise — an entity that allows managed Google Play to distribute apps to devices.

Enterprise Display Name

Business Name*

Contact Info

Contact Email

Data Protection Officer

Name

Email

Phone

EU Representative

Name

Email

Phone

□ I have read and agree to the Managed Google Play agreement.

Note: Please request and upload token to register Enterprise

Register

Note: The Register button gets enabled only after uploading *(on page 30)* the token file received from the HCL BigFix MCM Admin .

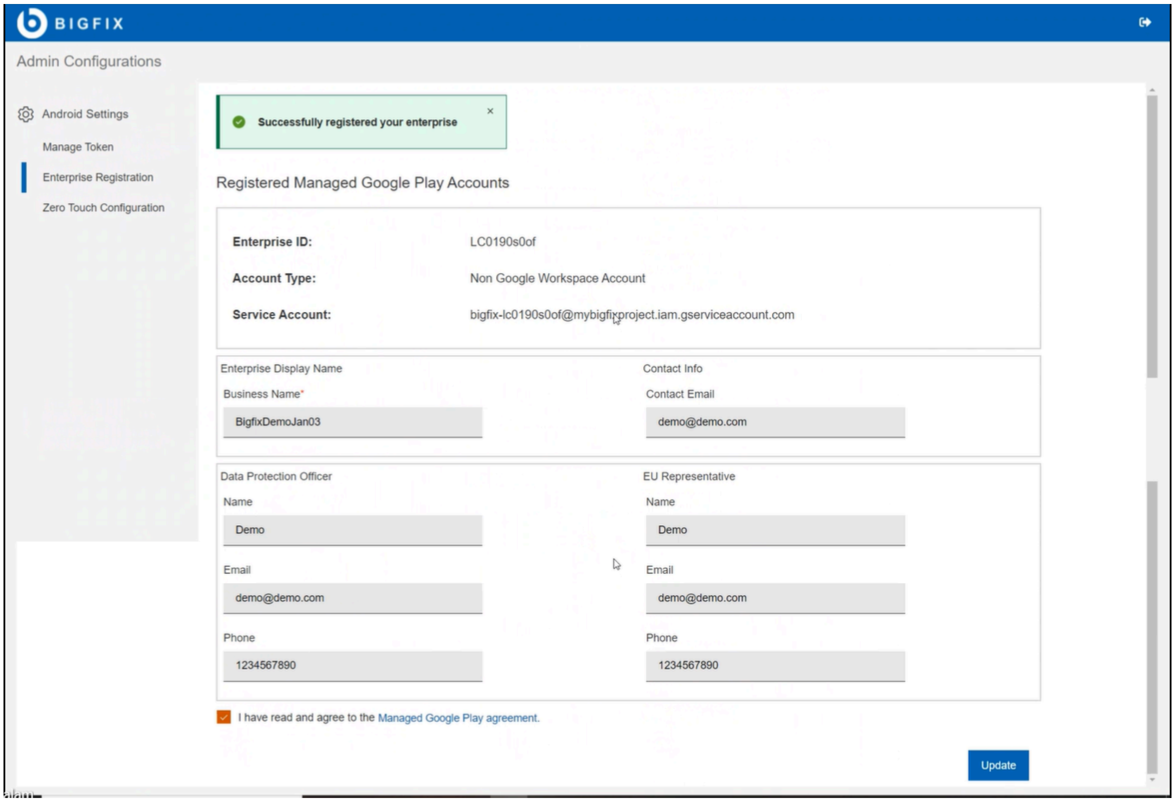
a. **Business Name**: This field is required. Enter the name of the business to be displayed. The number of characters must be less than 100.
b. **Data Protection Officer**: This is optional. Enter the Name, Email, and Phone number of the person responsible for the data.
c. **EU Representative**: This is optional. Enter the Name, Email, and Phone number of the person who represents the enterprise.
d. **Contact Info**: This is optional. Enter the email of the person to contact in the enterprise.

> **Note:** After the enterprise is registered, you can modify the details of Data Protection Officer, EU Representative, and the Contact Info.

e. Select the consent check box for Managed Google Play agreement.
f. Click **Register**.

A service account is created that is uniquely identifiable by business name and is auto-provisioned.

After completing the registration, BigFix MCM Admin Configuration page displays the enterprise ID, Account Type, Service Account, and all the other optional information entered.



Successfully registered your enterprise

Registered Managed Google Play Accounts

Enterprise ID: LC0190s0of

Account Type: Non Google Workspace Account

Service Account: bigfix-lc0190s0of@mybigfixproject.iam.gserviceaccount.com

Enterprise Display Name

Business Name*: BigfixDemoJan03

Contact Info

Contact Email: demo@demo.com

Data Protection Officer

Name: Demo

Email: demo@demo.com

Phone: 1234567890

EU Representative

Name: Demo

Email: demo@demo.com

Phone: 1234567890

☑ I have read and agree to the Managed Google Play agreement.

This process creates the Enterprise service account, service account credentials, and encrypts the credentials and saves to certificates directory. IT admins can now provision Android devices to the created enterprise account. Managed Google Play Accounts (user accounts) are automatically created when devices are provisioned.

3. Update information: You can update the optional enterprise information such as the Data Protection Officer details, EU Representative details, and the Contact Email if required. To do that:
    a. Update the required information.
    b. Select the consent check box for Managed Google Play agreement.
    c. Click **Update**.

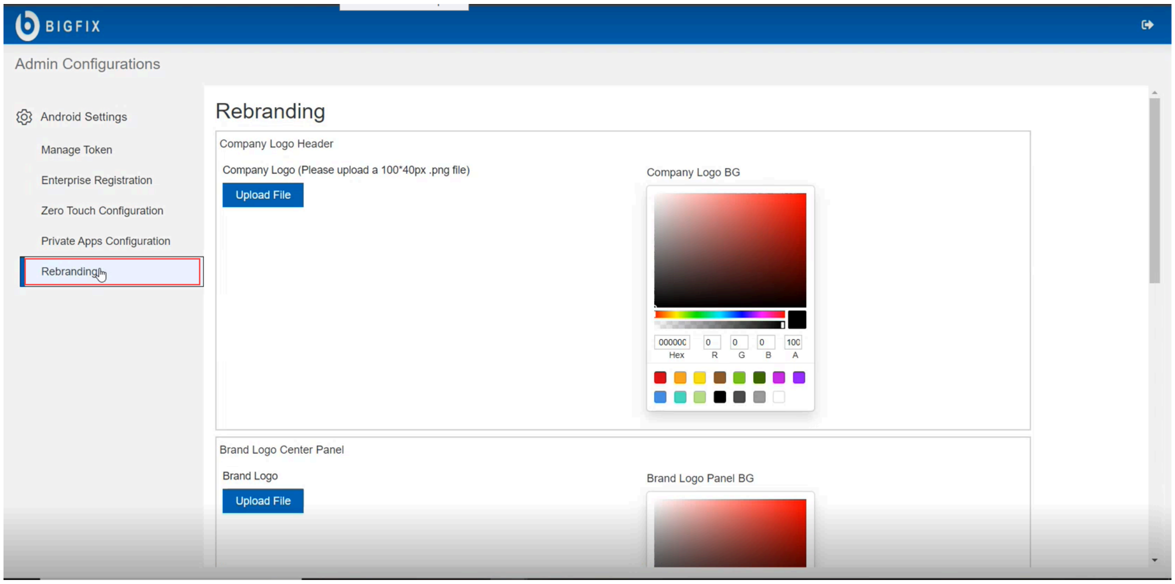
Rebranding user interfaces

You can change the appearance of the Enrollment UI and Android Server Configuration UI by changing the color, image, logo, brand name and so on.

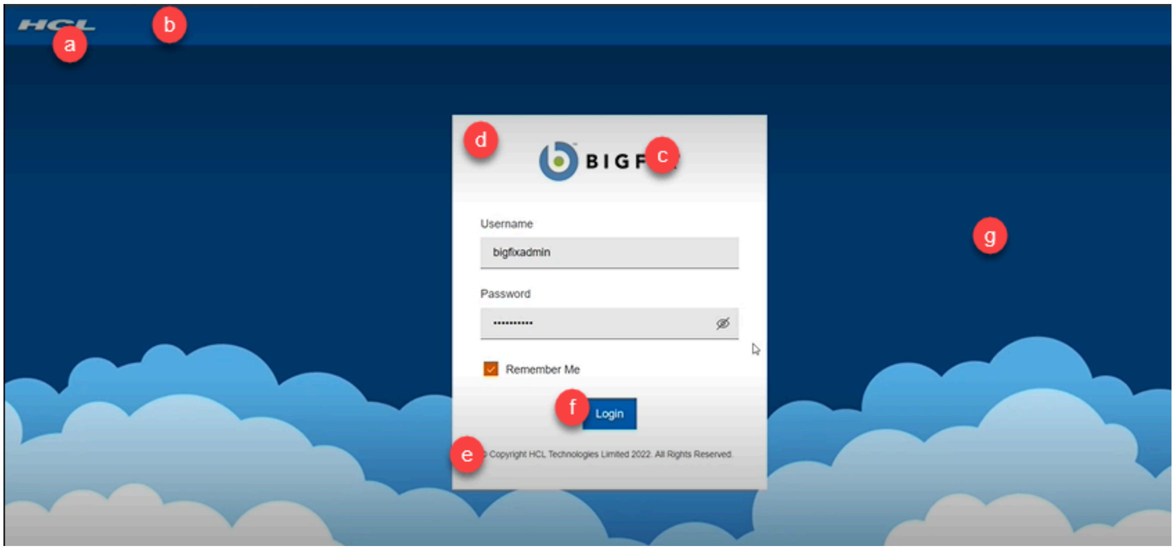
Rebranding Android Server Configuration UI

To change the appearance of the UI, do the following:

1. Log in to the Android Server Admin Configuration page with valid credentials.
2. From the left pane click **Rebranding**.



3. Change the configuration as desired.

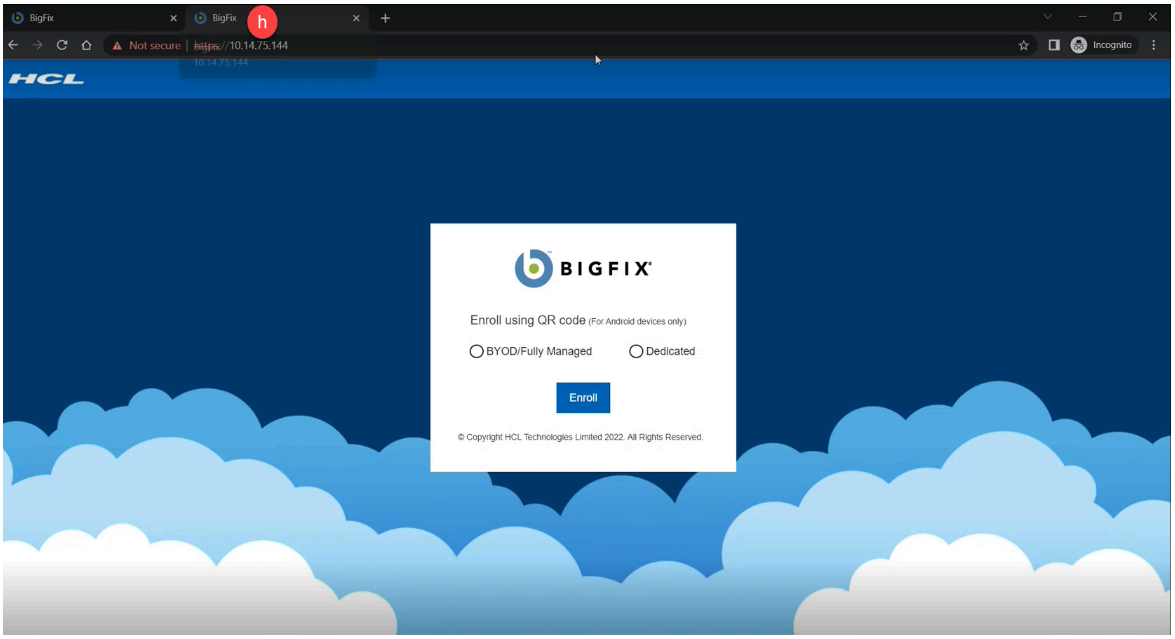


Username: bigfixadmin

Password: ••••••••••

☑ Remember Me

© Copyright HCL Technologies Limited 2022. All Rights Reserved.



Enroll using QR code (For Android devices only)

○ BYOD/Fully Managed ○ Dedicated

© Copyright HCL Technologies Limited 2022. All Rights Reserved.

a. To change the company logo, under Company Logo Header section, click **Upload File** and select the preferred logo file.
b. To change the background color of the header where the company logo is present, from the color panel in the Company Logo Header section, select a color.
c. To change the brand logo, under Brand Logo, click **Upload File** and select the preferred logo file.
d. To change the background color of the brand logo panel, from the color panel in the Brand Logo section, select a color.
e. To change the Copyright statement, click the Copyright statement and enter the desired text.
f. To change the button color, from the color panel in the Button Color section, select a color.
g. To change the image in the page background, under Page Background, click **Upload File** and select the preferred image file.

   h. To change the title of the browser tab, from the Browser Tab section, under **Favicon Title** enter the desired text.
   i. To change the icon in the browser tab, under Favicon image, click **Upload File** and select a preferred icon file.
4. Click **Configure**.
5. A message pops up to indicate to re-login to apply the new configuration. Click **Ok** to proceed.
6. Re-login to see the configuration changes.
7. To reset the configuration changes to the initial default settings, click **Reset**.

Note: The UI configuration changes are preserved even after the MDM server is restarted.

Uninstall MDM components

Learn how to uninstall MDM components.

Before you begin: You must be a Master Operator to perform this task through WebUI.

Uninstall MDM server

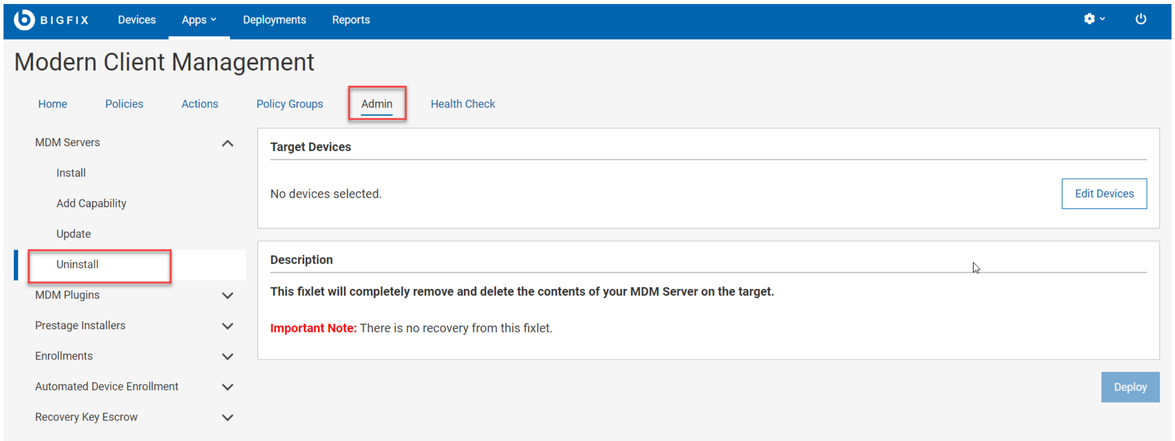
Uninstalling MDM server removes BigFix MDM from the server and you cannot use MDM services any longer from that server. There is no recovery from an MDM Server Uninstall. For the MDM devices to enroll and properly report again, MDM must be reinstalled.

To uninstall MDM server:

1. From the WebUI main page, click **Apps > MCM**.

2. On the Modern Client Management page, click **Admin**.

3. On the Admin page, from the left navigation, under MDM Server, click **Uninstall**



Target Devices

No devices selected.

Description

This fixlet will completely remove and delete the contents of your MDM Server on the target.

Important Note: There is no recovery from this fixlet.

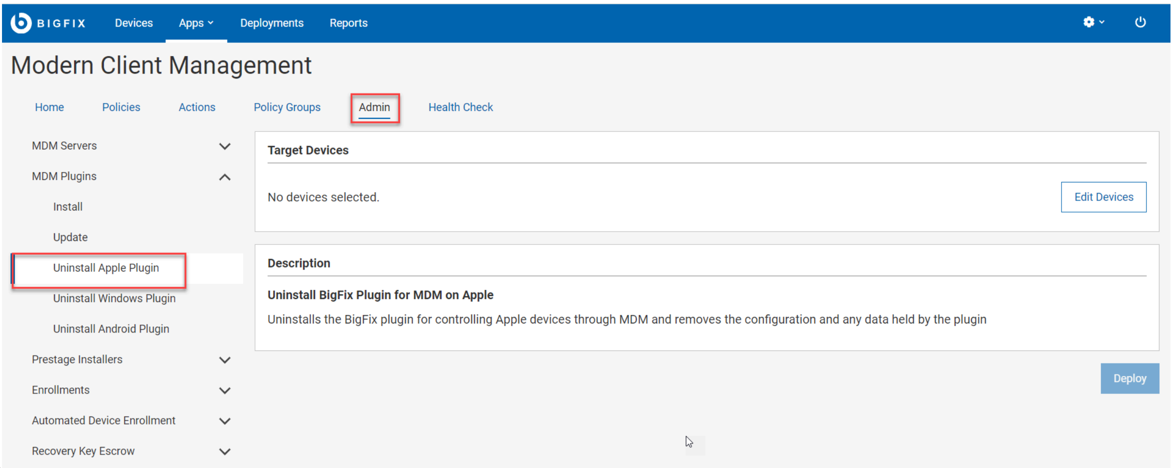
4. Click **Edit Devices** and select the MDM server that you want to uninstall.
5. Click **Deploy**.

Uninstall MDM Plugin for Apple

After uninstalling MDM Plugin for Apple from a device, you cannot manage Apple devices from that server.

To uninstall:

1. From the WebUI main page, click **Apps > MCM**.
2. On the Modern Client Management page, click **Admin**
3. On the Modern Client Management page, from the left pane under MDM Plugins, click **Uninstall Apple Plugin**.



No devices selected.

Uninstall BigFix Plugin for MDM on Apple

Uninstalls the BigFix plugin for controlling Apple devices through MDM and removes the configuration and any data held by the plugin

4. Click **Edit Devices** and select the server you want to uninstall the MDM plugin.
5. Click **Deploy**.

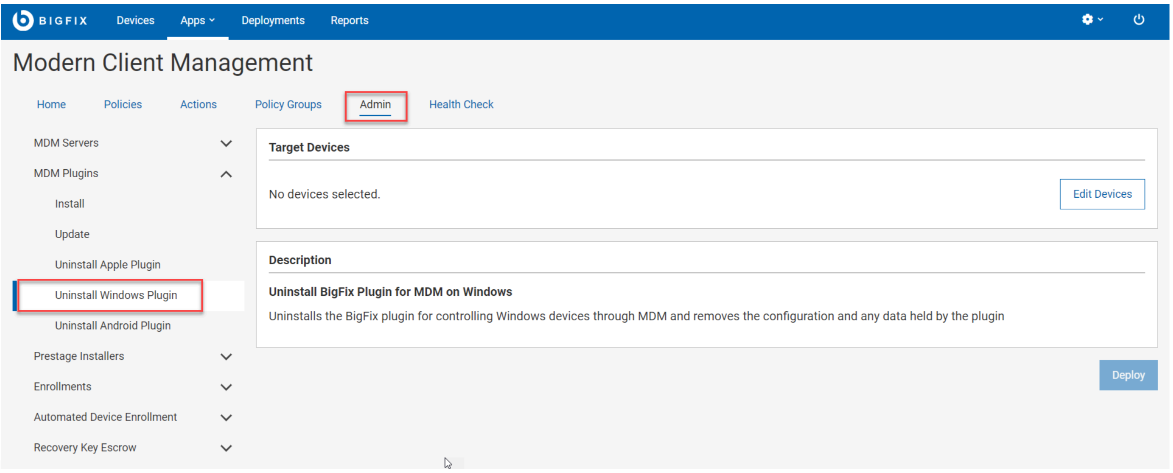
Uninstall MDM Plugin for Windows

After uninstalling MDM plugin for Windows from a device, you cannot manage Windows devices from that Plugin Portal.

To uninstall:

1. From the WebUI main page, click **Apps > MCM**.
2. On the Modern Client Management page, click **Admin**.

3. On the Modern Client Management page, from the left pane under MDM Plugins, click **Uninstall Windows Plugin**



Target Devices

No devices selected.

Description

Uninstall BigFix Plugin for MDM on Windows

Uninstalls the BigFix plugin for controlling Windows devices through MDM and removes the configuration and any data held by the plugin

4. Click **Edit Devices** and select the devices you want to uninstall Windows MDM plugin.

5. Click **Deploy**.

Uninstall MDM Plugin for Android

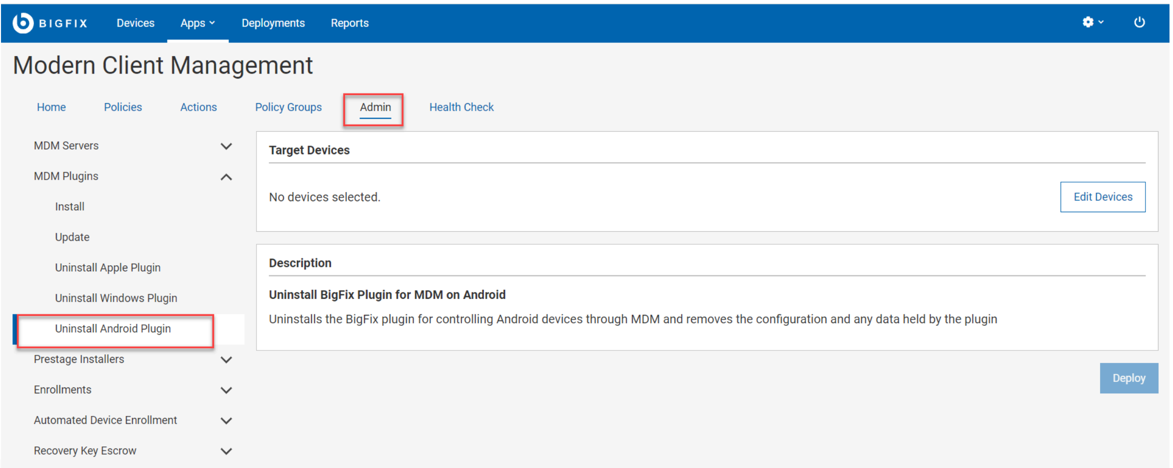
After uninstalling MDM Plugin for Android from a device, you cannot manage Android devices from that Plugin Portal.

To uninstall:

1. From the WebUI main page, click **Apps > MCM**.

2. On the Modern Client Management page, click **Admin**.

3. On the Modern Client Management page, from the left pane under MDM Plugins, click **Uninstall Android Plugin**



Target Devices

No devices selected.

Description

Uninstall BigFix Plugin for MDM on Android

Uninstalls the BigFix plugin for controlling Android devices through MDM and removes the configuration and any data held by the plugin

4. Click **Edit Devices** and select the devices you want to uninstall Android MDM Plugin.

5. Click **Deploy**.

Related reference

Error while reinstalling MDM server *(on page 62)*

BigFix MDM Server TLS Certificate Content

Understand the required format of the BigFix MDM Server TLS certificate for MDM Server installation.

BigFix MDM server TLS Certificate Content

The MDM Server certificate must be available in a `.crt` or `.pem` format, and must take the form of a certificate chain containing the following:

- The actual MDM TLS certificate provided by the trusted CA
- Any intermediate certificates provided by the trusted CA
- The trusted CA root certificate

If the trusted CA does not provide such a chain directly, concatenate the individual `.crt` or `.pem` files into a single certificate chain and provide it as the MDM Server's TLS certificate during MDM Server installation.

The following command is an example for concatenating certificates on Linux:

```
cat <server TLS .crt> [intermediate .crt] <CA root .crt> > mdmserver.crt
```

This may require additional action on one or more files provided by a trusted CA to extract the various certificates and keys needed to build the required chain.

Wildcard certificates

You can use wildcard TLS certificates for your MDM Server in several scenarios.

For example:

- when your MDM server is configured in cloud infrastructure where a load balancer terminates TLS.
- when your MDM Server is terminating TLS from devices on the Internet through a firewall hole.

In these cases, you can obtain a wildcard certificate from your Trusted CA of choice (example of the form "*.test.bigfix.com") instead of a specific FQDN like "mdm.test.bigfix.com".

Having a wildcard certificate allows separate servers in the domain to be able to share the same TLS certificate. For example, you can have two separate MDM servers: mdm1.test.bigfix.com and mdm2.test.bigfix.com, and both can be handled through the same wildcard certificate.

With whichever option you use, the wildcard certificate must be configured on the server that is actually terminating TLS from MDM enrolling devices.

- In cloud infrastructure, this would likely be a front-end load balancer which terminates TLS and routes connections to a cloud-based MDM Server VM. This load balancer will likely have a cloud-specific FQDN by default, so a CNAME must be defined in your DNS server to route your devices to correct TLS termination point using the desired FQDN. For example, entering a URL of https://mdm.test.bigfix.com would resolve to a load balancer which could have a name like "mdm-test.bigfix-com-1216115951.eu-central-1.elb.amazonaws.com".
- For a standard on-premises solution where the MDM server is running in the DMZ and there is a firewall hole to let port 443 traffic in to reach the MDM server, the MDM server must be set to use the wildcard certificate.

Just as with a dedicated certificate specifying the FQDN of a specific server, you must provide the certificate chain and not just the individual TLS certificate when supplying the wildcard certificate. Example:

- *.test.bigfix.com
- Intermediate CA supplied by the Trusted CA
- Root CA supplied by the Trusted CA

Related reference

BigFix MDM Server TLS Certificate Content *(on page 40)*

Related information

BigFix PlugIn and MDM SSL certificates and keys *(on page 11)*

Supported system environments

This section provides information about the supported system environments for MCM.

Component	**Supported environment**
Plugin Portal and Plugin	• RHEL7.4+ • RHEL8+ • Windows 2012 R2+
MDM Server Host and Docker	RHEL 7 to RHEL 8
Device Operating System	Windows 10 and Windows 11 (Pro, Enterprise, and Home*), macOS 10.14 and later, Android 10.0 and later, iOS 14 and later, iPadOS 14 and later

* Only certain Windows editions support all available operating system features that are configured through MDM. For complete information, see the Windows Configuration service provider reference document. Each CSP highlights which Windows Editions are supported.

Component	Supported environment
Docker Engine	• CE v19.x • RHEL version 1.13 or later
Docker-compose	1.25.x
MongoDB	• RHEL7.4+ • RHEL8+ • Windows 10 • Windows 11 • Windows Server 2016+

Troubleshooting

This section is intended to help you solve problems that might occur when installing BigFix MCM and BigFix Mobile.

Logging

The MCM component generates log files which can provide extra information when you troubleshoot an issue.

Log file locations

The following table shows the location of various MCM logs that are stored in your Windows and Linux systems.

Component	Windows	Linux
Plugin Portal log (Configurable through _BESPluginPortal_HTTPServer _LogFilePath)	`C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\BESPluginPortal.log`	`/var/log/BESPluginPortal.log`
Windows MDM Plugin	`C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins\WindowsMDMPlugin\Logs`	`/var/opt/BESPluginPortal/Plugins/WindowsMDMPlugin/Logs`
Apple MDM Plugin	`C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins\AppleMDMPlugin\Logs`	`/var/opt/BESPluginPortal/Plugins/AppleMDMPlugin/Logs`
Windows MDM Server	N/A	`/var/opt/BESUEM/windows/logs/windowsmdm.log`

Component	Windows	Linux
Apple MDM Server	N/A	`/var/opt/BESUEM/apple/logs/micromdm.log`
Apple MDM Gateway	N/A	`/var/opt/BESUEM/apple/logs/mdmgateway.log`
Apple MDM Webhook	N/A	`/var/opt/BESUEM/apple/logs/mdmwebhook.log`
Android MDM Plugin	`C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins\AndroidMDMPlugin\Logs`	`/var/opt/BESPluginPortal/Plugins/AndroidMDMPlugin/Logs`
Android MDM Server	N/A	`/var/opt/BESUEM/windows/logs/androidmdm.log`
WebUI log (Configurable through the server setting _WebUI_Logging_LogPath)	`c:\Program Files (x86)\BigFix Enterprise\BESWebUI\WebUI\logs\`	`/var/opt/BESWebUI/WebUI/logs/`

Client settings - Verbose Logs for Plugin or Plugin Portal

Parameter	Description
`_BESPluginPortal_Log_Verbose`	Sets the Plugin Portal Verbose to Off if the value is 0 (only critical messages are logged); sets it to On (to enable more logging) if the value is 1. The default value is 0.

`_BESPluginPortal_Log_EnabledLogs`	When Plugin Portal Verbose is enabled, sets the Plugin Portal message type configuration. Possible values include: all, critical, debug, timing, events. You can add values delimited by semicolons.
	Example:
	`_BESPluginPortal_Log_Verbose = 1`
	`_BESPluginPortal_Log_EnabledLogs = events;timing`
	The default message type is “'all”.
	Note: This can have a detrimental impact on Plugin Portal performance.

Parameter	Description
`_WindowsMDMPlugin_LogVerbose`	Sets the Windows MDM Plugin Verbose On if the value is 1 and sets Off if the value is 0. The default value is 0.
`_AppleMDMPlugin_LogVerbose`	Sets the Apple Plugin Verbose On if the value is 1 and sets Off if the value is 0. The default value is 0.
`_AndroidMDMPlugin_LogVerbose`	Sets the Android Plugin Verbose On if the value is 1 and sets Off if the value is 0. The default value is 0.

Logrotate

Logrotate handles the automatic rotation and compression of log files to manage available disk space. On the 0th minute of every hour, log files are rotated, and old log files are compressed and kept as backup by running cron jobs in the following containers:

- windowsmdm
- applemdm
- rabbitmq
- openresty

The backup file name format is <Filename.log>_<YYYY-MM-DD>_<HH-mm-ss>.gz. For example, `mdmgateway.log_2020-06-03_07-13-00.gz`

Logs are rotated under the following conditions and backed up to the following log locations.

Container	Log location	Max File size	Rotate Count	Scheduled cron JOB running
For all containers	Container's internal path	10 MB	10	every hour 0th minute
For macmdm log	/var/opt/BESUEM/mac/logs	50 MB	10	every hour 0th minute
For windowsmdm log	/var/opt/BESUEM/windows/logs	50 MB	10	every hour 0th minute
For openresty log	/var/opt/BESUEM/openresty/logs	50 MB	10	every hour 0th minute

Note: If log file size is less than 50 MB, running the logrotate cron job does not create a new log file.

MCM Debug Logging

Debug logs can help you troubleshoot issues because these logs have an extended logging level. In the BESUEM site, look for Fixlets that are marked as TROUBLESHOOTING to enable logging on different MDM components; they are relevant if you have the components available.

Related information

MDM debug tool *(on page 60)*

Windows error codes

The following includes commonly encountered error messages and related information.

Windows Endpoint enrollment error codes

Error code	Description	Cause
0x80190191	Unauthorized user	User is not authorized.
0x80190190	The server could not process the transfer request.	The syntax of the remote file name is invalid.
0x801901F4	It prevents the apps from opening or forces them to close soon after opening.	Incorrectly configured system settings or irregular entries in the Windows registry.
0x8600023	Already Imported this Package	Importing this PPKG has been attempted before, and the action failed
0x80180026	Device is ExternallyManaged	This occurs when a device is locked in ProvisioningMode.
0x80192ee7	Network Name Not resolved	Either DNS is not available or your computer does not have internet access.
0x8004002	File Not Found	The Provisioning Package file cannot be read.
0x8004005	Access Denied	This occurs when UAC is disabled, or when someone clicks ‘No’ at the MDM enrollment prompt.

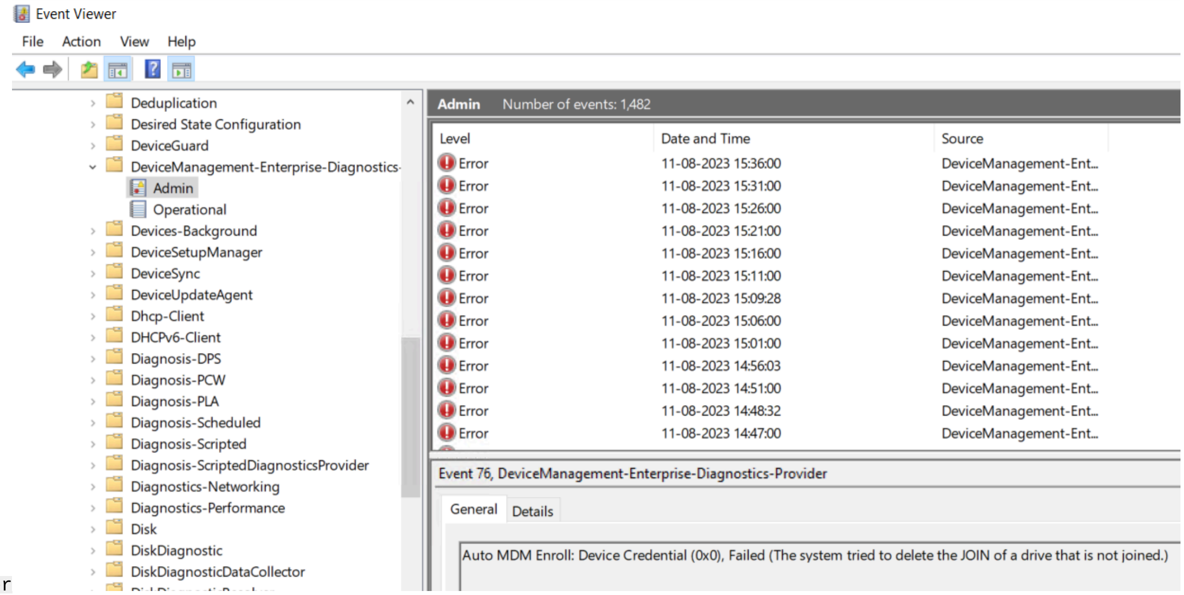
For a comprehensive list of error codes, see https://docs.microsoft.com/en-us/windows/win32/mdmreg/mdm-registration-constants.

Event Viewer log

If there are any issues in the end point MDM commands or policies, and if the MDM logs do not have enough log information, you can check Windows Event Viewer log for possible reasons.

To access Event Viewer:

1. From the MDM enrolled Windows endpoint, open the Start Menu and in the search bar, type "Event Viewer" to find and open the Event Viewer app.
2. From the Event Viewer, navigate to `Event Viewer > Applications and Services Logs > Microsoft > Windows > Device Management-Enterprise-Diagnostics-Provider`



Admin Number of events: 1,482

Level	Date and Time	Source
Error	11-08-2023 15:36:00	DeviceManagement-Ent...
Error	11-08-2023 15:31:00	DeviceManagement-Ent...
Error	11-08-2023 15:26:00	DeviceManagement-Ent...
Error	11-08-2023 15:21:00	DeviceManagement-Ent...
Error	11-08-2023 15:16:00	DeviceManagement-Ent...
Error	11-08-2023 15:11:00	DeviceManagement-Ent...
Error	11-08-2023 15:09:28	DeviceManagement-Ent...
Error	11-08-2023 15:06:00	DeviceManagement-Ent...
Error	11-08-2023 15:01:00	DeviceManagement-Ent...
Error	11-08-2023 14:56:03	DeviceManagement-Ent...
Error	11-08-2023 14:51:00	DeviceManagement-Ent...
Error	11-08-2023 14:48:32	DeviceManagement-Ent...
Error	11-08-2023 14:47:00	DeviceManagement-Ent...

Event 76, DeviceManagement-Enterprise-Diagnostics-Provider

Auto MDM Enroll: Device Credential (0x0), Failed (The system tried to delete the JOIN of a drive that is not joined.)

Installing Docker CE and Docker compose on RHEL8

Install Docker CE and Docker compose on RHEL 8

To install Docker CE and Docker compose on RHEL 8:

1. Add the external repository by running the following command.

   ```
   sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
   ```

   a. Verify whether the repository has been enabled. To do that, run the following command that returns detailed information about all the enabled repositories.

   ```
   sudo dnf repolist -v
   ```

2. Install docker-ce with the --nobest option. With this option, the first version of `docker-ce` with satisfiable dependencies is selected as the "fallback" version.

   ```
   sudo dnf install --nobest docker-ce
   ```

3. Install the latest available containerd.io package manually

   ```
   sudo dnf install
    https://download.docker.com/linux/centos/7/x86_64/stable/Packages/containerd.io-1.2.6-3.3.el7.x86_64.
   rpm
   ```

4. Install the latest docker-ce version:

   ```
   sudo dnf install docker-ce
   ```

5. Start and enable the docker daemon

   ```
   sudo systemctl enable --now docker
   ```

   a. Confirm whether the daemon is active by running this command:

      ```
      systemctl is-active docker
      ```

6. Install docker-compose globally.

   a. Download the binary file from the project’s GitHub page:

      ```
      curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname
       -s)-$(uname -m)" -o docker-compose
      ```

   b. After the binary file is downloaded, move it to the `/usr/local/bin` folder, and then make it executable:

      ```
      sudo mv docker-compose /usr/local/bin && sudo chmod +x /usr/local/bin/docker-compose
      sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
      ```

For detailed information, see https://linuxconfig.org/how-to-install-docker-in-rhel-8

After this installation, you might encounter a Docker CE container connectivity issue. Complete these steps to resolve this issue.

Resolve Docker CE container connectivity issue

To resolve Docker CE container connectivity issue:

1. Check which interface Docker is using. For example, 'docker0'.

   ```
   ip link show
   ```

2. Check available firewalld zones. For example, 'public'

   ```
   sudo firewall-cmd --get-active-zones
   ```

3. Check which zone the Docker interface is bound to. Typically, the Docker interface is not bound to a zone yet.

   ```
   sudo firewall-cmd --get-zone-of-interface=docker0
   ```

4. Add the 'docker0' interface to the 'public' zone. Changes are visible only after the firewalld is reloaded

```
sudo nmcli connection modify docker0 connection.zone public
```

5. Masquerading enables Docker ingress and egress.

```
sudo firewall-cmd --zone=public --add-masquerade --permanent
```

6. Reload the firewalld

```
sudo firewall-cmd --reload
```

7. Restart dockerd

```
sudo systemctl restart docker
```

Troubleshooting LDAPS connection

Condition

LDAP connection failure.

Cause

It is optional to configure MDM server with LDAP credentials. If you enter wrong values or values in incorrect format, it displays an error as "invalid". However, the Fixlet actions complete successfully, which might cause connection issues at times.

Solution

Using the command-line utility BESmdmldaputil, you can validate LDAP parameters, email, and user authentication to troubleshoot your LDAP connection issues.

Note: If you change LDAP parameters in `.env` file, you must restart `idservice` for the changes to take effect.

To validate LDAP parameters, run the following command from the MDM server:

```
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil <options>
```

where the options include the following:

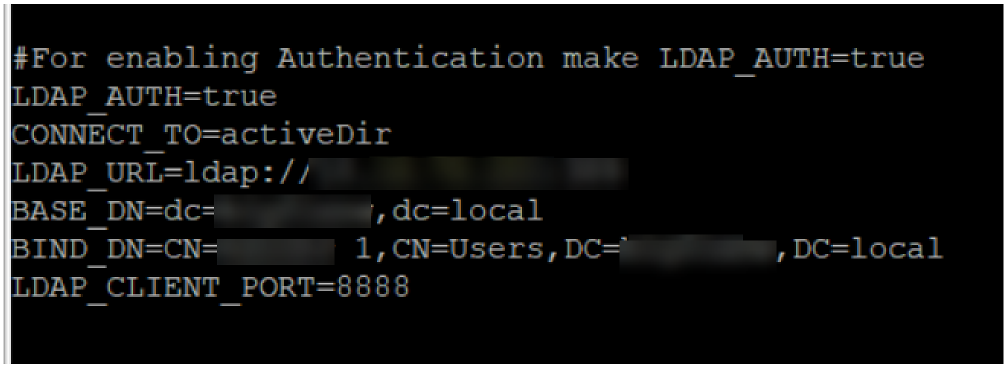
```
-a : Authenticate user
-c : Clear cache
-e : Validate email
-f : Get all AD/AAD groups
-g : Get group list
-h : Help Content
-l : List cache names
-p : Get attribute list
```

```
-u : Get user configuration
-v : Validate .env variables, values, and AD/Azure AD connectivity
```

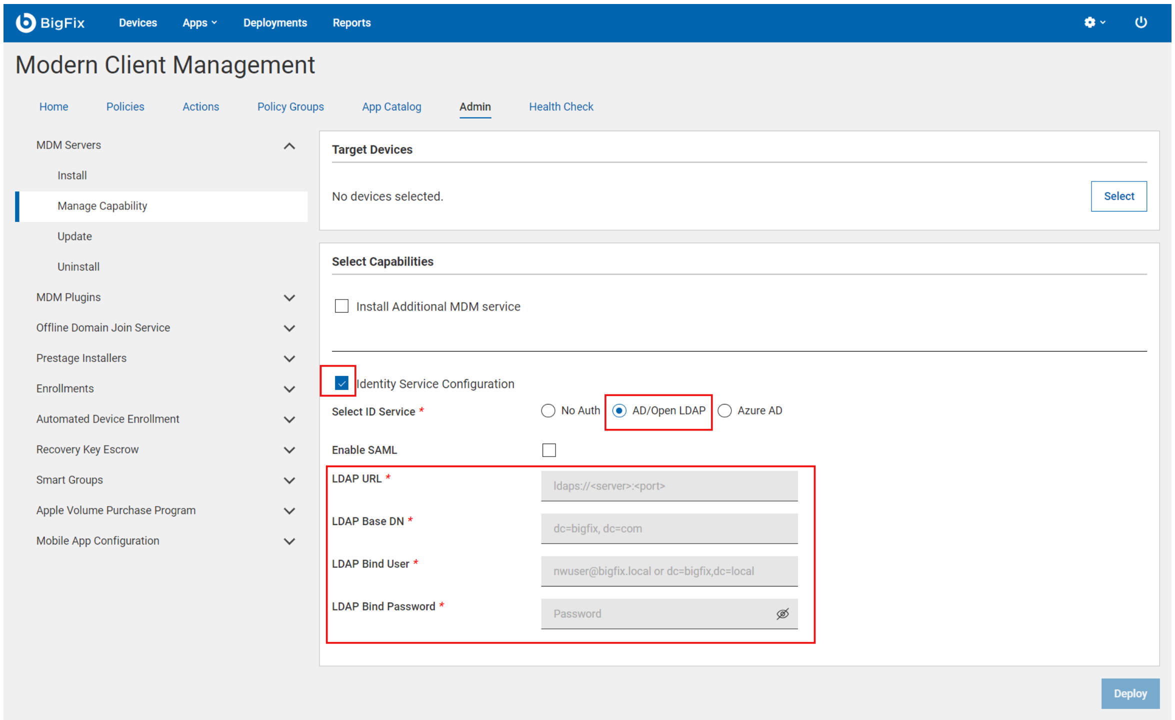
The following are some of the examples on how to use the options;

```
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -h
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -v
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -e user@example.com
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -a username:password
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -e user@example.com -a username:password
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -c groupNames
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -l
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -f
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -p
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -u user@example.com
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -g
```

- This utility validates LDAP parameters that were provided through WebUI Identity Service Configuration. as shown in the following image:



```
#For enabling Authentication make LDAP_AUTH=true
LDAP_AUTH=true
CONNECT_TO=activeDir
LDAP_URL=ldap://[redacted]
BASE_DN=dc=[redacted],dc=local
BIND_DN=CN=[redacted] 1,CN=Users,DC=[redacted],DC=local
LDAP_CLIENT_PORT=8888
```



Modern Client Management

Target Devices

No devices selected.

Select Capabilities

□ Install Additional MDM service

☑ Identity Service Configuration

Select ID Service * ○ No Auth ● AD/Open LDAP ○ Azure AD

Enable SAML □

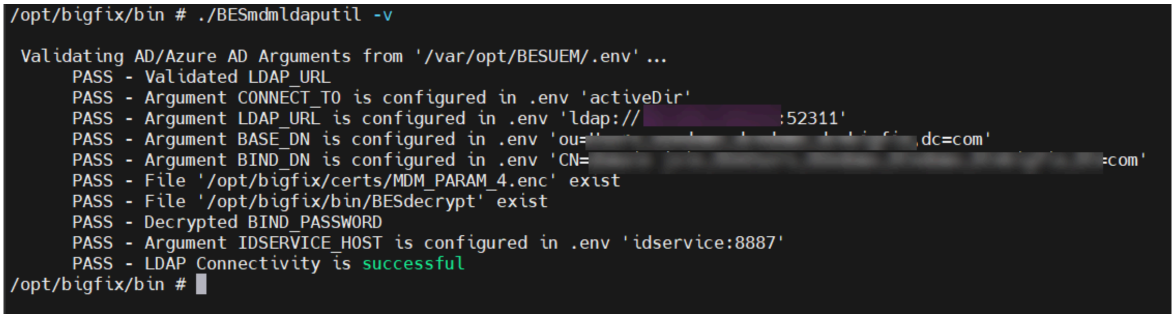
LDAP URL * ldaps://<server>:<port>

LDAP Base DN * dc=bigfix, dc=com

LDAP Bind User * nwuser@bigfix.local or dc=bigfix,dc=local

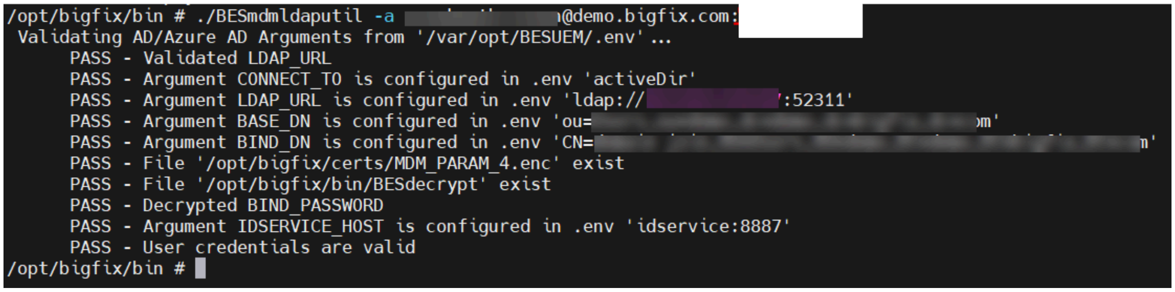
LDAP Bind Password * Password

- The following is an example of the validation messages when you use the option -v to validate LDAP env arguments.



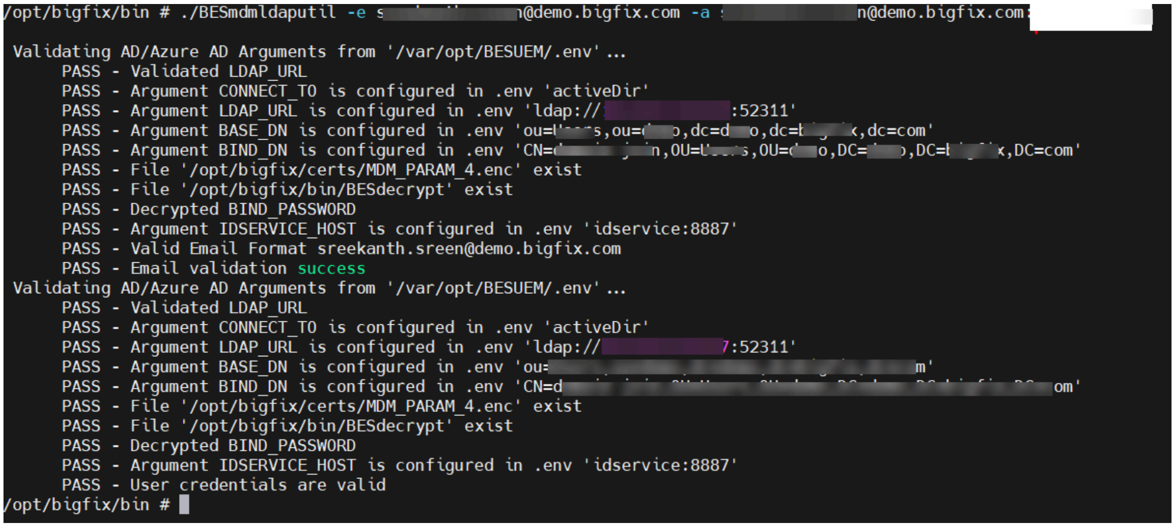
```
/opt/bigfix/bin # ./BESmdmldaputil -v

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env' ...
      PASS - Validated LDAP_URL
      PASS - Argument CONNECT_TO is configured in .env 'activeDir'
      PASS - Argument LDAP_URL is configured in .env 'ldap://[redacted]:52311'
      PASS - Argument BASE_DN is configured in .env 'ou=[redacted],dc=com'
      PASS - Argument BIND_DN is configured in .env 'CN=[redacted]=com'
      PASS - File '/opt/bigfix/certs/MDM_PARAM_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - LDAP Connectivity is successful
/opt/bigfix/bin #
```

- The following is an example of the validation messages when you use the option -a to authenticate a specific user.



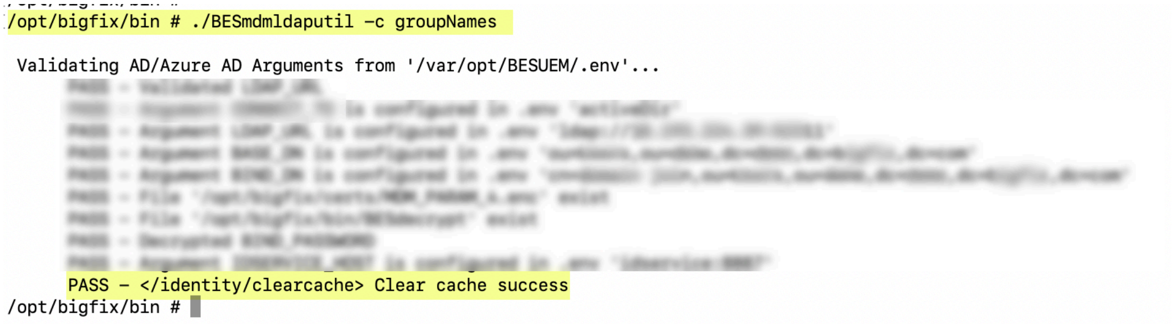
```
/opt/bigfix/bin # ./BESmdmldaputil -a [redacted]@demo.bigfix.com:[redacted]
 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env' ...
      PASS - Validated LDAP_URL
      PASS - Argument CONNECT_TO is configured in .env 'activeDir'
      PASS - Argument LDAP_URL is configured in .env 'ldap://[redacted]:52311'
      PASS - Argument BASE_DN is configured in .env 'ou=[redacted]om'
      PASS - Argument BIND_DN is configured in .env 'CN=[redacted]m'
      PASS - File '/opt/bigfix/certs/MDM_PARAM_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - User credentials are valid
/opt/bigfix/bin #
```

- You can also combine more than one option to get the desired result. The following image shows the result for the options -e and -a for the values provided:



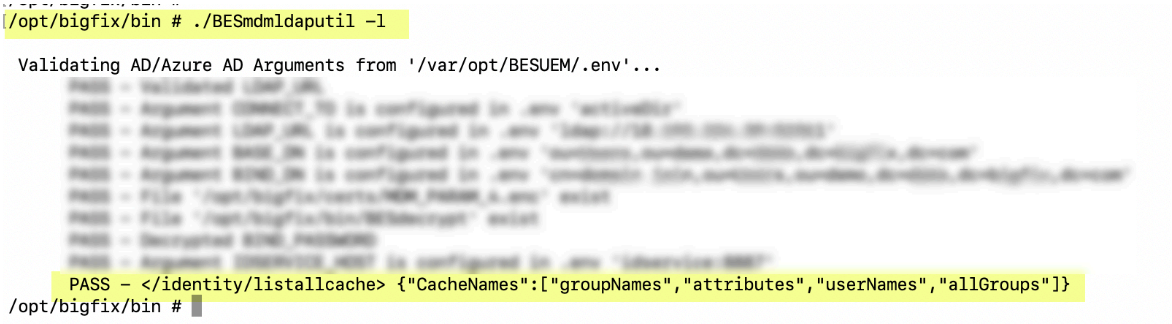
```
/opt/bigfix/bin # ./BESmdmldaputil -e s[redacted]@demo.bigfix.com -a [redacted]n@demo.bigfix.com

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env' ...
      PASS - Validated LDAP_URL
      PASS - Argument CONNECT_TO is configured in .env 'activeDir'
      PASS - Argument LDAP_URL is configured in .env 'ldap://[redacted]:52311'
      PASS - Argument BASE_DN is configured in .env 'ou=[redacted],ou=[redacted],dc=[redacted],dc=[redacted],dc=com'
      PASS - Argument BIND_DN is configured in .env 'CN=[redacted],OU=[redacted],OU=[redacted],DC=[redacted],DC=[redacted],DC=com'
      PASS - File '/opt/bigfix/certs/MDM_PARAM_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - Valid Email Format sreekanth.sreen@demo.bigfix.com
      PASS - Email validation success
 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env' ...
      PASS - Validated LDAP_URL
      PASS - Argument CONNECT_TO is configured in .env 'activeDir'
      PASS - Argument LDAP_URL is configured in .env 'ldap://[redacted]7:52311'
      PASS - Argument BASE_DN is configured in .env 'ou=[redacted]m'
      PASS - Argument BIND_DN is configured in .env 'CN=d[redacted]om'
      PASS - File '/opt/bigfix/certs/MDM_PARAM_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - User credentials are valid
/opt/bigfix/bin #
```

- The following is an example to clear cache with the option -c.



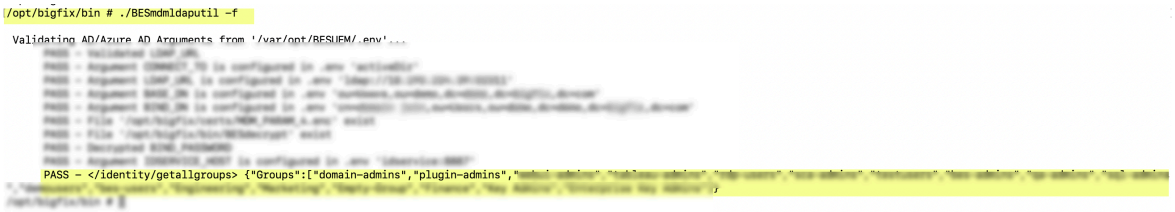
```
/opt/bigfix/bin # ./BESmdmldaputil –c groupNames

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS – </identity/clearcache> Clear cache success
/opt/bigfix/bin #
```

- The following is an example to list cache names with the option -l.



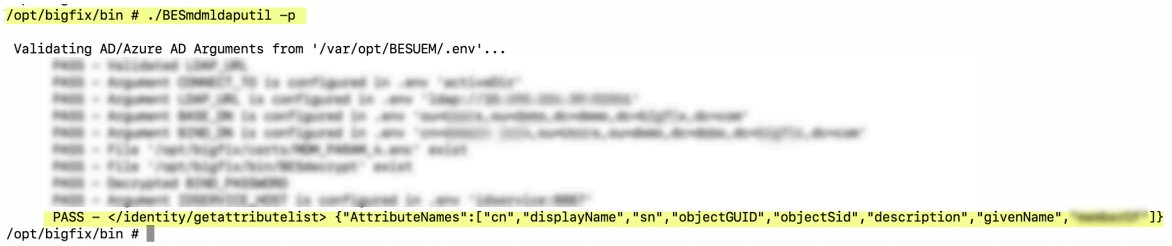
```
/opt/bigfix/bin # ./BESmdmldaputil –l

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS – </identity/listallcache> {"CacheNames":["groupNames","attributes","userNames","allGroups"]}
/opt/bigfix/bin #
```

- The following is an example to list all group names with the option -f.



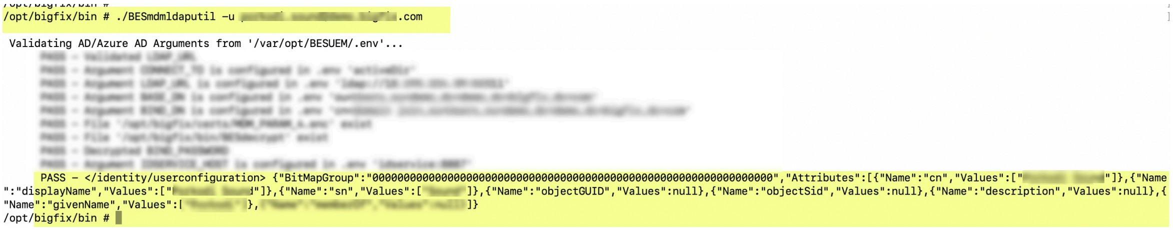
```
/opt/bigfix/bin # ./BESmdmldaputil -f

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/getallgroups> {"Groups":["domain-admins","plugin-admins","[redacted]
[redacted]
```

- The following is an example to list attributes names with the option -p.



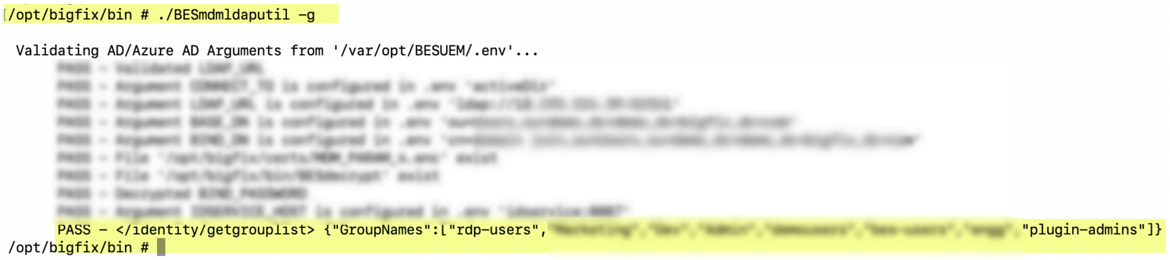
```
/opt/bigfix/bin # ./BESmdmldaputil -p

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/getattributelist> {"AttributeNames":["cn","displayName","sn","objectGUID","objectSid","description","givenName",[redacted]]}
/opt/bigfix/bin #
```

- The following is an example to get user configuration with the option -u.



```
/opt/bigfix/bin # ./BESmdmldaputil -u [redacted].com

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/userconfiguration> {"BitMapGroup":"0000000000000000000000000000000000000000000000000000000000000000","Attributes":[{"Name":"cn","Values":["[redacted]"]},{"Name
":"displayName","Values":["[redacted]"]},{"Name":"sn","Values":[[redacted]]},{"Name":"objectGUID","Values":null},{"Name":"objectSid","Values":null},{"Name":"description","Values":null},{
"Name":"givenName","Values":[[redacted]},[redacted]]}
/opt/bigfix/bin #
```

- The following is an example to get group names with the option -g.



```
/opt/bigfix/bin # ./BESmdmldaputil -g

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/getgrouplist> {"GroupNames":["rdp-users",[redacted]"plugin-admins"]}
/opt/bigfix/bin #
```

With this, you can understand if the configured connection is working, and if not, what specifically to look for.

Troubleshooting Azure connection

Condition

Azure connection failure.

Cause

It is optional to configure MDM server with Azure credentials. If you enter wrong values or values in incorrect format, it displays an error as “invalid”. However, the Fixlet actions complete successfully, which might cause connection issues at times.

Solution

Using the command-line utility BESmdmldaputil, you can validate Azure parameters, email, and user authentication to troubleshoot your Azure connection issues.

Note: If you change Azure parameters in `.env` file, you must restart `idservice` for the changes to take effect.

To validate Azure parameters, run the following command from the MDM server:

```
docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil <options>
```

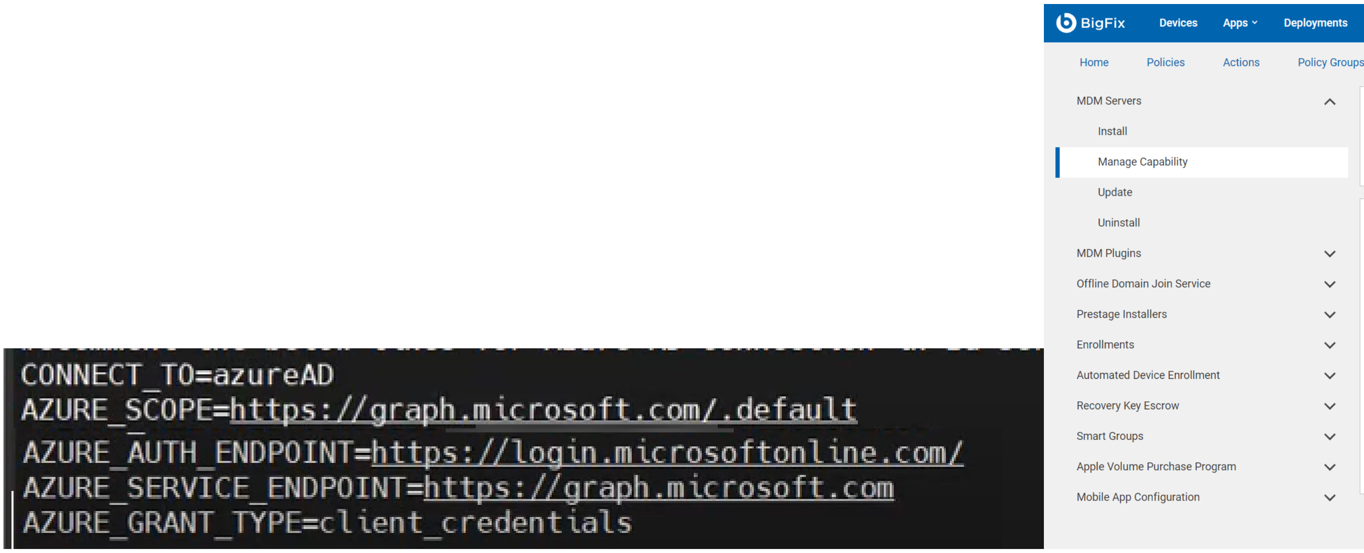
where the options include the following:

```
-a : Authenticate user
-c : Clear cache
-e : Validate email
-f : Get all AD/AAD groups
-g : Get group list
-h : Help Content
-l : List cache names
-p : Get attribute list
-u : Get user configuration
-v : Validate .env variables, values, and AD/Azure AD connectivity
```

The following are some of the examples on how to use the options;

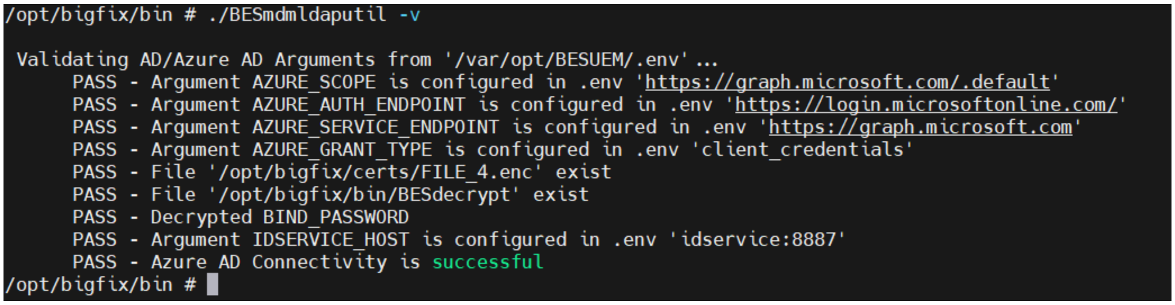
```
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -h
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -v
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -e user@example.com
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -a username:password
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -e user@example.com -a username:password
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -c groupNames
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -l
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -f
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -p
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -u user@example.com
 docker exec -it idservice /opt/bigfix/bin/BESmdmldaputil -g
```

- This utility validates Azure parameters that were provided through WebUI Identity Service Configuration as shown in the following image:



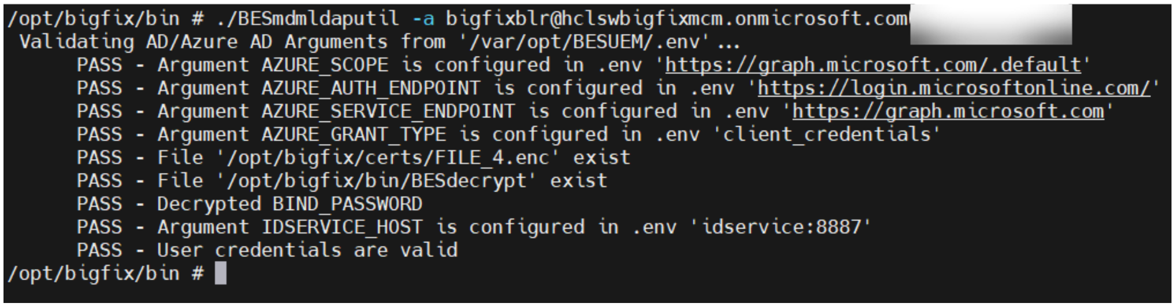
```
CONNECT_TO=azureAD
AZURE_SCOPE=https://graph.microsoft.com/.default
AZURE_AUTH_ENDPOINT=https://login.microsoftonline.com/
AZURE_SERVICE_ENDPOINT=https://graph.microsoft.com
AZURE_GRANT_TYPE=client_credentials
```

- The following is an example of the validation messages when you use the option -v to validate Azure env arguments.



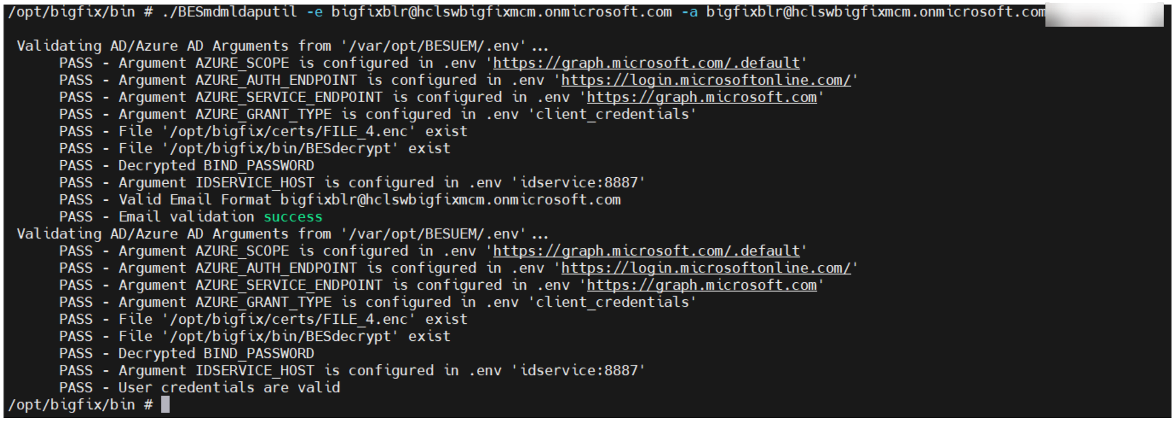
```
/opt/bigfix/bin # ./BESmdmldaputil -v

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env' ...
      PASS - Argument AZURE_SCOPE is configured in .env 'https://graph.microsoft.com/.default'
      PASS - Argument AZURE_AUTH_ENDPOINT is configured in .env 'https://login.microsoftonline.com/'
      PASS - Argument AZURE_SERVICE_ENDPOINT is configured in .env 'https://graph.microsoft.com'
      PASS - Argument AZURE_GRANT_TYPE is configured in .env 'client_credentials'
      PASS - File '/opt/bigfix/certs/FILE_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - Azure AD Connectivity is successful
/opt/bigfix/bin #
```

- The following is an example of the validation messages when you use the option -a to authenticate a specific user.



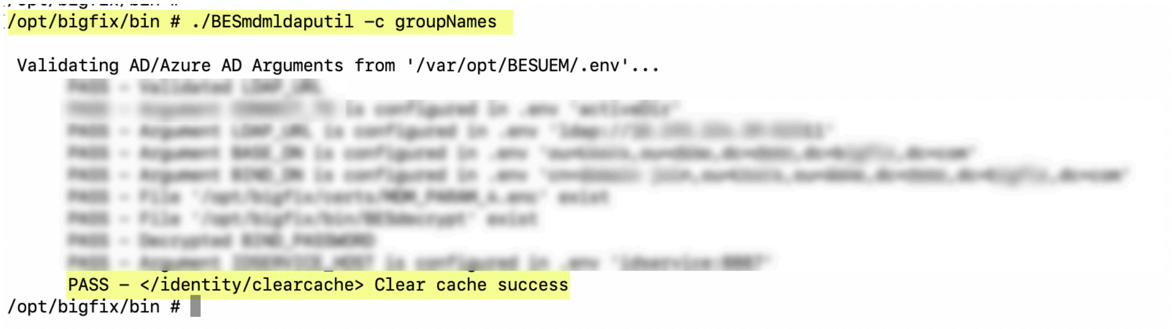
```
/opt/bigfix/bin # ./BESmdmldaputil -a bigfixblr@hclswbigfixmcm.onmicrosoft.com [redacted]
 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env' ...
      PASS - Argument AZURE_SCOPE is configured in .env 'https://graph.microsoft.com/.default'
      PASS - Argument AZURE_AUTH_ENDPOINT is configured in .env 'https://login.microsoftonline.com/'
      PASS - Argument AZURE_SERVICE_ENDPOINT is configured in .env 'https://graph.microsoft.com'
      PASS - Argument AZURE_GRANT_TYPE is configured in .env 'client_credentials'
      PASS - File '/opt/bigfix/certs/FILE_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - User credentials are valid
/opt/bigfix/bin #
```

- You can also combine more than one option to get the desired result. The following image shows the result for the options -e and -a for the values provided:



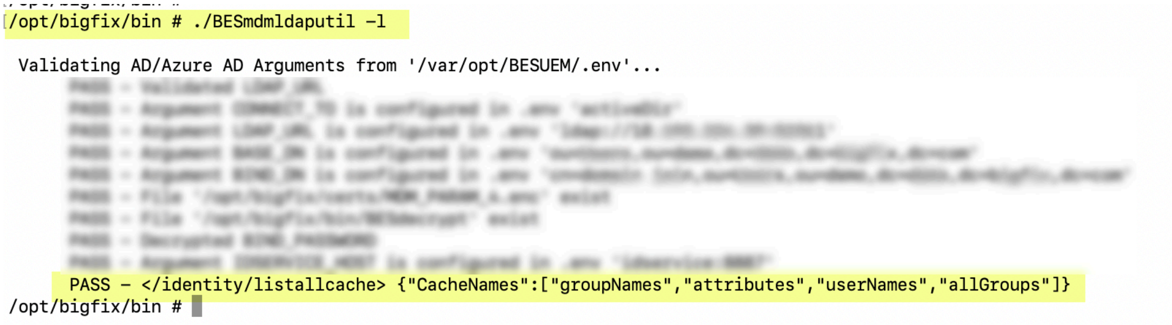
```
/opt/bigfix/bin # ./BESmdmldaputil -e bigfixblr@hclswbigfixmcm.onmicrosoft.com -a bigfixblr@hclswbigfixmcm.onmicrosoft.com[redacted]

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      PASS - Argument AZURE_SCOPE is configured in .env 'https://graph.microsoft.com/.default'
      PASS - Argument AZURE_AUTH_ENDPOINT is configured in .env 'https://login.microsoftonline.com/'
      PASS - Argument AZURE_SERVICE_ENDPOINT is configured in .env 'https://graph.microsoft.com'
      PASS - Argument AZURE_GRANT_TYPE is configured in .env 'client_credentials'
      PASS - File '/opt/bigfix/certs/FILE_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - Valid Email Format bigfixblr@hclswbigfixmcm.onmicrosoft.com
      PASS - Email validation success
 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      PASS - Argument AZURE_SCOPE is configured in .env 'https://graph.microsoft.com/.default'
      PASS - Argument AZURE_AUTH_ENDPOINT is configured in .env 'https://login.microsoftonline.com/'
      PASS - Argument AZURE_SERVICE_ENDPOINT is configured in .env 'https://graph.microsoft.com'
      PASS - Argument AZURE_GRANT_TYPE is configured in .env 'client_credentials'
      PASS - File '/opt/bigfix/certs/FILE_4.enc' exist
      PASS - File '/opt/bigfix/bin/BESdecrypt' exist
      PASS - Decrypted BIND_PASSWORD
      PASS - Argument IDSERVICE_HOST is configured in .env 'idservice:8887'
      PASS - User credentials are valid
/opt/bigfix/bin #
```

- The following is an example to clear cache with the option -c.



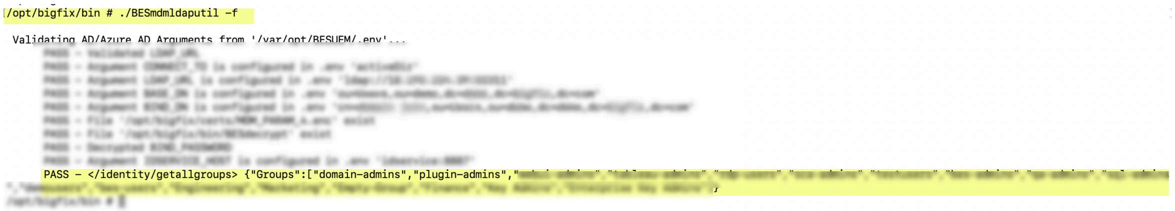
```
/opt/bigfix/bin # ./BESmdmldaputil -c groupNames

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/clearcache> Clear cache success
/opt/bigfix/bin #
```

- The following is an example to list cache names with the option -l.



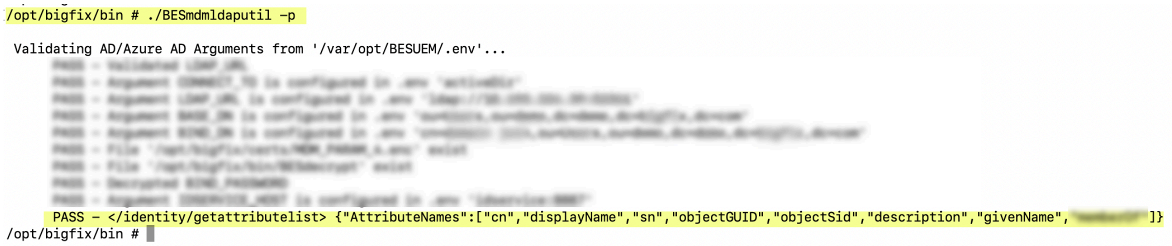
```
/opt/bigfix/bin # ./BESmdmldaputil -l

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/listallcache> {"CacheNames":["groupNames","attributes","userNames","allGroups"]}
/opt/bigfix/bin #
```

- The following is an example to list all group names with the option -f.



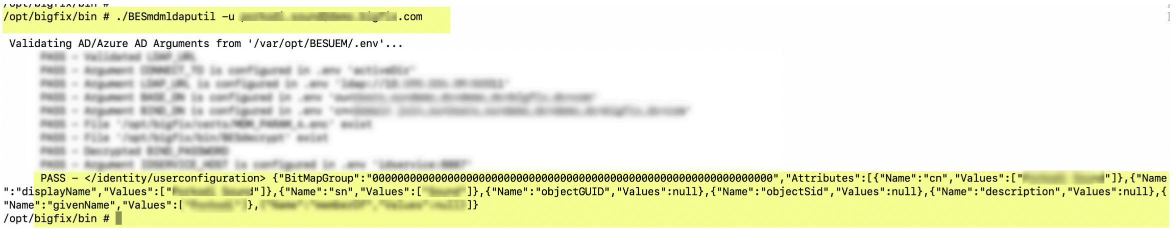
```
/opt/bigfix/bin # ./BESmdmldaputil -f

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
      [redacted]
      PASS - </identity/getallgroups> {"Groups":["domain-admins","plugin-admins","[redacted]
```

- The following is an example to list attributes names with the option -p.



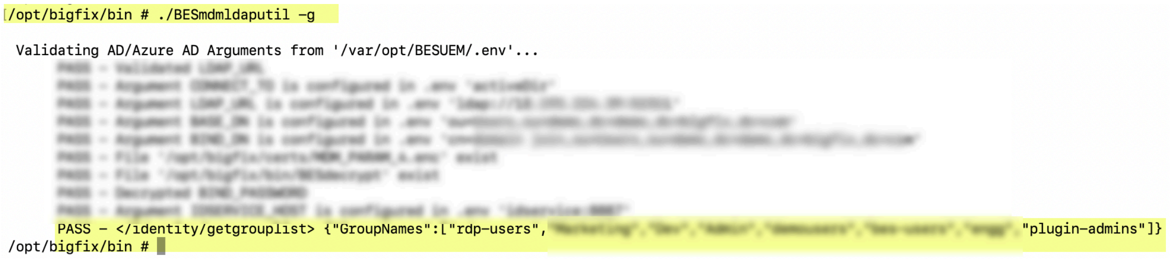
```
/opt/bigfix/bin # ./BESmdmldaputil -p

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
     [redacted]
     PASS - </identity/getattributelist> {"AttributeNames":["cn","displayName","sn","objectGUID","objectSid","description","givenName",[redacted]]}
/opt/bigfix/bin #
```

- The following is an example to get user configuration with the option -u.



```
/opt/bigfix/bin # ./BESmdmldaputil -u [redacted].com

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
     [redacted]
     PASS - </identity/userconfiguration> {"BitMapGroup":"0000000000000000000000000000000000000000000000000000000000000000000","Attributes":[{"Name":"cn","Values":["[redacted]"]},{"Name
":"displayName","Values":["[redacted]"]},{"Name":"sn","Values":[[redacted]]},{"Name":"objectGUID","Values":null},{"Name":"objectSid","Values":null},{"Name":"description","Values":null},{
"Name":"givenName","Values":[[redacted]},[redacted]]}
/opt/bigfix/bin #
```

- The following is an example to get group names with the option -g.



```
/opt/bigfix/bin # ./BESmdmldaputil -g

 Validating AD/Azure AD Arguments from '/var/opt/BESUEM/.env'...
     [redacted]
     PASS - </identity/getgrouplist> {"GroupNames":["rdp-users",[redacted]"plugin-admins"]}
/opt/bigfix/bin #
```

With this, you can understand if the configured connection is working, and if not, what specifically to look for.

DEP troubleshooting

You can find DEP enrollment troubleshooting information in this section.

Device to profile assignments

MicroMDM writes all DEP devices information to `/opt/bigfix/config/dep-devices.json` file. This file is refreshed approximately every 30 minutes. You can see the profile status and UUID of the profile along with the other device information. You can find out the current device to profile assignments of every enrolled device from this file.

Sample content of `dep-devices.json` file

```
[
   {
      "serial_number":"************",
      "model":"iPad",
      "description":"IPAD WI-FI 32GB SPACE GRAY-USA",
      "color":"SPACE GRAY",
      "asset_tag":"",
      "profile_status":"assigned",
      "profile_uuid":"180E3526801006EB204EDC9C3A4C3141",
      "DEPProfileAssignedDate":"2020-10-06T21:07:54Z"
   },
   {
      "serial_number":"************",
      "model":"MacBook Pro 15\"",
      "description":"MBP 15.4/16GB",
      "color":"SILVER",
      "asset_tag":"",
      "profile_status":"pushed",
      "profile_uuid":"180E3526801006EB204EDC9C3A4C3141",
      "DEPProfileAssignedDate":"2020-10-06T21:07:54Z"
   },
   {
      "serial_number":"************",
      "model":"iPhone XR",
      "description":"IPHONE XR BLACK 64GB VZW-USA",
      "color":"BLACK",
      "asset_tag":"",
      "profile_status":"assigned",
      "profile_uuid":"180E3526801006EB204EDC9C3A4C3141",
      "DEPProfileAssignedDate":"2020-10-06T21:07:54Z"
   }
]
```

Monitor MacOS MCM component logs and metrics

DEP logs are available in `/var/log/apple-mdm.log` file.

Sample content of `apple-mdm.log` file

```
{

  "cursor": "MDowOjE2MTA2NzY3NTA3NzM6MTYxMDY4MTUxOTA5NDp0cnVlOjE2MTA2NzY3NTA3NzM",
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/depsync.go:313",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.(*Watcher).Run",
  "level": "info",
  "module": "default",
  "msg": "Sync DEP devices",
  "time": "2021-01-15T04:01:58Z"
}
{
  "devices": 0,
  "fetched": "2021-01-15T02:12:30Z",
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/depsync.go:347",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.(*Watcher).Run",
  "level": "info",
  "module": "default",
  "more": false,
  "msg": "DEP sync success",
  "phase": "sync",
  "time": "2021-01-15T04:01:59Z"
}
{
  "device_count": 0,
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/device/worker.go:124",
  "func": "github.com/micromdm/micromdm/platform/device.(*Worker).updateFromDEPSync",
  "level": "info",
  "module": "default",
  "msg": "Updating devices from DEP",
  "time": "2021-01-15T04:02:00Z"
}
{
  "FilterUDID": null,
  "count of udids in query": 0,
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/device/builtin/db.go:65",
  "func": "github.com/micromdm/micromdm/platform/device/builtin.(*DB).List",
  "level": "info",
  "module": "restapi",
  "msg": "List device db operation called",
  "time": "2021-01-15T04:02:00Z"
}

{
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/define_profile.go:46",
  "func": "github.com/micromdm/micromdm/platform/dep.MakeDefineProfileEndpoint.func1",
  "level": "info",
  "module": "restapi",
  "msg": "Apply dep profile request",
  "profile_name": "DEPTestProfile",
  "time": "2021-01-15T04:12:33Z"
}
{
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/define_profile.go:51",
  "func": "github.com/micromdm/micromdm/platform/dep.MakeDefineProfileEndpoint.func1",
  "level": "info",
  "module": "restapi",
  "msg": "Apply dep profile success",
  "profile_name": "DEPTestProfile",
  "profile_uuid": "EB7A65BC96583B0C77DE990633C2EAD3",
  "time": "2021-01-15T04:12:33Z"
}
{
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/apply_autoassigner.go:32",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.MakeApplyAutoAssignerEndpoint.func1",
  "level": "info",
  "module": "restapi",
  "msg": "Apply dep autoassigner request",
  "profile_uuid": "EB7A65BC96583B0C77DE990633C2EAD3",
  "time": "2021-01-15T04:12:33Z"
}
{
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/apply_autoassigner.go:37",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.MakeApplyAutoAssignerEndpoint.func1",
  "level": "info",
  "module": "restapi",
  "msg": "Apply dep autoassigner success",
  "profile_uuid": "EB7A65BC96583B0C77DE990633C2EAD3",
  "time": "2021-01-15T04:12:33Z"
}
{
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/syncnow.go:21",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.MakeSyncNowEndpoint.func1",
  "level": "info",
  "module": "restapi",
  "msg": "Dep syncnow request",
  "time": "2021-01-15T04:13:01Z"
}
{
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/depsync.go:380",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.(*Watcher).Run",
  "level": "info",
  "module": "default",
  "msg": "Explicit DEP sync requested",
  "time": "2021-01-15T04:13:01Z"
}
```

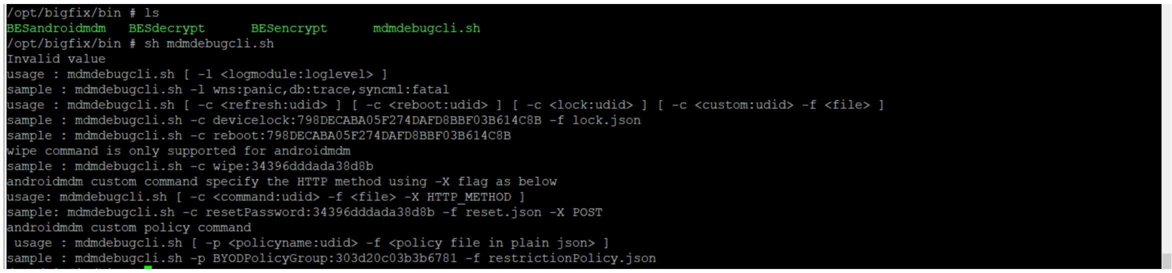
```
{
  "cursor": "MDowOjE2MTA2NzY3NTA3NzM6MTYxMDY4MzMxOTA5Mjp0cnVlOjE2MTA2NzY3NTA3NzM",
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/depsync.go:313",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.(*Watcher).Run",
  "level": "info",
  "module": "default",
  "msg": "Sync DEP devices",
  "time": "2021-01-15T04:13:01Z"
}
{
  "devices": 0,
  "fetched": "2021-01-15T02:12:30Z",
  "file": "/opt/bigfix/src/github.com/micromdm/micromdm/platform/dep/sync/depsync.go:347",
  "func": "github.com/micromdm/micromdm/platform/dep/sync.(*Watcher).Run",
  "level": "info",
  "module": "default",
  "more": false,
  "msg": "DEP sync success",
  "phase": "sync",
  "time": "2021-01-15T04:13:01Z"
}
```

MDM debug tool

This command-line tool can be used to set log levels for individual/group/all MDM modules, execute commands and update policy settings on the MDM enrolled devices using REST APIs. This will be helpful to quickly debug production issues when there is a communication failure at different MDM layers and to trace the execution work flows, requests, responses from/to end points.

To use this tool do the following:

1. Login to MDM Windows/Android containers using the command `docker exec –it <windowsmdm or androidmdm> sh`
2. Run the command `/opt/bigfix/bin/mdmdebugcli.sh` to see the help information of the tool as follows:



```
/opt/bigfix/bin # ls
BESandroidmdm   BESdecrypt      BESencrypt      mdmdebugcli.sh
/opt/bigfix/bin # sh mdmdebugcli.sh
Invalid value
usage : mdmdebugcli.sh [ -l <logmodule:loglevel> ]
sample : mdmdebugcli.sh -l wns:panic,db:trace,syncml:fatal
usage : mdmdebugcli.sh [ -c <refresh:udid> ] [ -c <reboot:udid> ] [ -c <lock:udid> ] [ -c <custom:udid> -f <file> ]
sample : mdmdebugcli.sh -c devicelock:798DECABA05F274DAFD8BBF03B614C8B -f lock.json
sample : mdmdebugcli.sh -c reboot:798DECABA05F274DAFD8BBF03B614C8B
wipe command is only supported for androidmdm
sample : mdmdebugcli.sh -c wipe:34396dddada38d8b
androidmdm custom command specify the HTTP method using -X flag as below
usage: mdmdebugcli.sh [ -c <command:udid> -f <file> -X HTTP_METHOD ]
sample: mdmdebugcli.sh -c resetPassword:34396dddada38d8b -f reset.json -X POST
androidmdm custom policy command
 usage : mdmdebugcli.sh [ -p <policyname:udid> -f <policy file in plain json> ]
sample : mdmdebugcli.sh -p BYODPolicyGroup:303d20c03b3b6781 -f restrictionPolicy.json
```

Find managed configuration properties of an Android app

You can also use MDM debug tool to Find managed configuration properties of an Android app. To do that follow these steps:

1. Login to MDM Android containers using the command `docker exec –it <androidmdm> sh`
2. Run the command `/opt/bigfix/bin/mdmdebugcli.sh [-c applications:packageName]` where packageName is the Bundle ID of an application. For example, to find the managed configuration properties of Microsoft Outlook, enter the following command:

   ```
   /opt/bigfix/bin/mdmdebugcli.sh -c applications:com.microsoft.office.outlook
   ```

   . You get the response in Base64 encoded format.
3. Decode the response to get the list of managed configuration parameters.

Ignore MDM server vulnerability due to TLS 1.0

Read this section to address MDM Server security exposure.

Problem

For versions up to MCM 2.1, vulnerability scan on the MDM Server detects exposure due to MDM Server accepting connections using TLS 1.0 and TLS 1.1.

Cause

This vulnerability exposure is specific to the port 5671 only. Up to MCM 2.1, port 5671 uses TLS 1.0 for internal communication.

The encryptions through TLS 1.0 was formally deprecated in March 2021 due to security issues. Websites using TLS 1.0 are considered non-compliant by PCI since 30 June 2018. PCI Data Security Standard (PCI DSS) does not consider TLS 1.0 to be strong enough to protect sensitive information transferred to or from web sites.

Therefore, the vulnerability scan detects the exposure.

It does not impact the ports 443 or 8443 as TLS V1.2 or higher is forced for communications through these ports.

Solution

You can safely ignore the vulnerability alert regarding the use of TLS 1.0.

This is because this vulnerability impacts only port 5671, which is used only for communication between RabbitMQ and the MDM Plugins internally. This port is not exposed to the Internet. This connection is controlled by client/server certificates, and therefore, only the MDM Plugins with those specific client/server certificates can establish a connection to initiate internal communication. Without appropriate client certificates, even the internal communication cannot be established.

This will not be an issue in versions later than MCM 2.1, as TLS V1.2 or higher is forced for communications through all the ports, and hence will be completely TLS requirement compliant even for internal communication.

Error while reinstalling MDM server

Read this page to troubleshoot the error while reinstalling an MDM server after uninstalling the same.

Problem

After uninstalling an MDM server from a RHEL VM, when a user tries to reinstall the same MDM server again, the WebUI throws the following error `Something went wrong, the specified server already has keys uploaded` and the WebUI action does not start.

Cause

If the credentials of the uninstalled MDM server are not removed automatically as expected, when the user tries to reinstall the same MDM server, this error is shown.

Solution

As a workaround, to fix this issue, after uninstalling MDM Server, complete the following steps:

1. From WebUI, go to **MCM > Admin**.
2. From the left navigation pane, explore **MDM Plugins** and select **Remove Credentials**.
3. From the Remove Credentials drop down, select the MDM Server for which you want to remove the credentials.
4. Click **Delete**.

Installing BigFix

To install the BigFix Platform, obtain a license, and run the installation wizard that guides you through the installation of the BigFix root server, console, client, and WebUI.

Purchase a license and obtain a BigFix license authorization file (`*.BESLicenseAuthorization`) by using your License Key Center account. In the case of a Proof-of-concept evaluation, contact your HCL Technical Sales Representative.

For details on installing BigFix and its components, see BigFix Installation.

Notices

This information was developed for products and services offered in the US.

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

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

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel

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

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel

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

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

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

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

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

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel

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

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

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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

Statements regarding HCL's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from HCL Ltd. Sample Programs.

Trademarks

HCL Technologies Ltd. and HCL Technologies Ltd. logo, and hcl.com are trademarks or registered trademarks of HCL Technologies Ltd., registered in many jurisdictions worldwide.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

Other product and service names might be trademarks of HCL or other companies.

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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