Mobile Center Administrator Best Practices

Document release date: October 2018

INSTALLATION AND CONFIGURATION

Mobile Center can be installed as a full installation (where there is no previous installation of Mobile Center) or an upgrade on top of an existing installation.

The installer checks what files are already installed and installs or updates the relevant files.

General deployment considerations

Mobile Center supports a distributed architecture where different test clients can interact with the same Mobile Center server instance.
Mobile Center deployment components are:

Component	Function

Mobile Center Server	This is a single web server that can be installed on a physical or virtual environment that:
	• mediates between the testing-tool client calls to mobile devices and provides a user interface within the testing tool for recording and running tests on real mobile devices.
	• accepts apps for testing and manages app versions.
	• provides a user interface (Lab Management console) for administrators to:
	  ○ manage users
	  ○ manage apps and view their properties such as OS and version
	  ▪ control devices: restart, unlock, or open a device remotely
	  ▪ view and manage connectors
	  ▪ configure a variety of settings for users such as proxy definitions and packaging services
	  ▪ enable extended services such as security scans, production usage, user sentiment, crowd testing, and SDK compliance.

Component	Function
	Note: When you install the Mobile Center server, it is also includes an embedded connector.
PostgreSQL database	You can choose to connect Mobile Center to an existing external PostgreSQL database or use database that is embedded into the Mobile Center Server installation (physical or virtual). You specify this option during installation. For details, see Mobile Center - Windows Installation or Mobile Center - Linux Installation.
Connector	The connector is designed as a lightweight piece of software for connecting devices to Mobile Center that can also be installed as a standalone component. You can install the connector on multiple machines in distributed locations, or on your testing-tool machine. The connector can be installed on a Windows, Linux, or Mac machine. The connector manages the physical USB connection to the device, and the logical state machine on top of it. The connector can be installed on virtual environment; however, it must maintain USB connectivity to the devices (USB Passthrough for mobile devices).

Deployment scenarios

The decision point for Mobile Center deployment scenario varies based on the customer's needs.

Scenario	Description	Advantages
All-in-one	Single box deployment for MC server, DB and embedded connector	Simplicity. Ideal for Proof of Concept and local installations.

Scenario	Description	Advantages
3-Tier Deployment	Separate Web and Data layers by installing MC server and DB on different locations	Scalability of Web and DB layers. Supports local IT best practices for Web and DB layer management.

For the connectors/device's deployment, the following scenarios can be considered:

Scenario	Description	Advantages
Central device hub	A central lab of devices connected to the connector on the Mobile Center server machine.	Efficiency. Avoids duplication of tasks for setting up and managing devices.
Distributed device hubs	Connectors installed on machines in multiple locations (on-site/off-site/globally dispersed).	Scalable. New labs can be added as needed.
Bring your own device	Connector installed on a developer's/testing engineer's machine.	Supports hands-on testing of the app on the device.

Hardware requirements

The full list of HW Requirements for Mobile Center is available at https://admhelp.microfocus.com/mobilecenter/en/latest/Content/Before_starting_installation.htm

The following parameters should be taken into consideration when planning Mobile Center hardware resources:

Component	Memory	CPU	Disk Space
Mobile Center Server	MC server is a Java application. Hence, it uses a predefined amount of host memory. The amount of consumed memory is impacted by the number of simulations session (user sessions). The minimal memory requirement is 4GB. For medium deployment (<30 devices) 8GB is recommended, and for large (>30 devices) 16GB.	MC server CPU consumption is dependent on the number of requests that are processed. The minimal requirement is an x64 processor, 2.2 GHz.	The disk space usage on the MC server depends on the number of logs generated, packaged applications, processes, etc. The recommendation is to have at least 20GB: 15GB for general installation and 5GB for TMP/TEMP folder
PostgreSQL DB	PostgreSQL memory consumption is impacted by SQL queries that it is required to execute. Minimal requirement for memory is 2GB. However, it is strongly recommended to have at least 8GB there for medium deployment (<30 devices) and 16GB (>30 devices overall).	PostgreSQL is process based. The minimal requirement is a dual-core CPU, 2.2 GHz.	The disk space usage on PostgreSQL is dependent on the data size. Mobile Center uploads AUTs (Application Under Test) into the database, so the data folder size will increase. On Windows, PostgreSQL is installed on the C: drive, so disk space must be allocated there.
Connector	MC Connector is a Java application. Hence, it uses a predefined amount of host memory.	The same guidelines as MC Server apply to the Connector Java application. Remote access to the device increases the	The disk space usage on the MC Connector depends on the number of logs generated, application

Component	Memory	CPU	Disk Space
	The amount of consumed memory is impacted by amount of simulation session (user sessions). The minimal requirement is 2GB. For standard deployments (8-10 devices per connector), it is recommended to have at least 8GB, and for large deployments (12-25 devices) 16GB.	CPU consumption and must be considered. The connector hardware must be planned according to the expected concurrent sessions on mobile devices. It differs slightly between Windows, Linux, and Mac connectors. But the “rule of thumb” is to allocate ½ CPU Core for each remote device session.	files cached on the connector, etc. The recommendation is to have at least 10GB.

Network Requirements

Mobile Center provides straight-forward network requirements.
You can find full information about MC architecture at
https://admhelp.microfocus.com/mobilecenter/en/latest/Content/HPMC_architecture.htm

Network Latency

Mobile Center is designed for resiliency over the network (WAN), by using REST API communication over the HTTP/S protocol. However, there is also communication channel that leverages the WebSocket protocol. Communication through this protocol may present some limitations that needs to be considered.
In general, with a network latency <100ms, there should be no communication issues with MC and connectors, using the public Internet, MPLS, VPN or any other method. A latency >200ms will introduce connectivity challenges.
To work on a device in remote view, the recommended network bandwidth is 1Mbps or higher.

Mobile Center and SSL

By default, MC uses an SSL configuration to communicate between a server and connectors It is achieved by generating a self-signed SSL certificates during the installation. For production usage, it is highly recommended to use CA certificates (certificate issued by Certification

Authority as opposed to self-signed). This will remove security warnings in browsers as well as streamline connectivity of testing tools. Also, it recommended to use a CA certificate together with a CA Root certificate, to avoid any recognition issues on the client machine. For more information, see https://admhelp.microfocus.com/mobilecenter/en/Latest/Content/SSL.htm
The use of SSL is also recommended from a networking perspective, in order to eliminate any internal security blockages by IPS or other security gateways.

Mobile Center Ports

Mobile Center Server (Web front end) utilizes a single port. The port is configured during the installation of MC Server.
The MC connector also utilizes a single port for connectivity with the MC server and end-user (client). Internally, the MC connector utilizes a reverse-proxy (Nginx) to route the requests to relevant mobile devices.
So, from the networking perspective, a single port for MC Server and Connector should be accessible (ingress).
Regarding used protocols, there is a requirement for HTTP/S and WebSocket/WebSocket Secure (WS/S) protocols.

Client tools and Mobile Center server connectivity

Common client tools are UFT, LoadRunner, Sprinter, BPM, LeanFT, and Appium scripts.
Testing-tool clients connect to the Mobile Center server for the following:

- A user interface (UI) for managing devices and uploading apps over HTTP/HTTPS.
- API (JSON commands) for tests and management, sent over WebSocket (WS/S).
- The remote screen viewer client sent over WebSocket (WS/S)

Connector Scalability

The Connector machine is capable of handling a significant number of mobile devices.
However, the upper limit of devices per single connector is defined by several parameters: operating system, motherboard hardware, USB ports and their versions, etc.
For example, Windows 7 has limitations relating to USB3.0 ports (supported natively in Windows 8), etc.
Therefore, the recommendations are as follows:

- Avoid using Windows 7 for an MC Connector machine (not part of supported configuration)
- Connect up to 20 mobile devices per single connector (using USB hubs – see the section below)
- For iOS-based deployments, consider using the MC Connector for OSX.
- For Android-based deployment, consider using the MC Connector for Linux or Windows (Android device drivers should be installed separately).
- Ensure that the OS is not configured for a hibernate or sleep, in order that the devices will remain stable there

USB Hubs and devices power consumption

While devices are used with Mobile Center, there is a need for synchronization and charging. The device is connected via a USB cable, so, it is constantly being charged, and the communication is done over it (MC Connector to Agent).

In order to support the required scalability and since the amount of USB ports are usually limited, use a “USB self-powered hub” a hub that takes its power from an external power supply unit and can therefore provide full power to every port.

Charging requirements for Mobile devices vary from 500 to 2100mA (from Android and iOS phones to tables and iPads). It is strongly recommended that you ensure that the power hub is capable of delivering the required amount to all USB ports.

For example: A power 7-usb ports hub that of 60W has a spec of 12V and 5A (12x5=60). The 5A will be split between 7 ports (a smart hub can do that dynamically), which will give ~714mA per port, which is sufficient for most of the mobile phones. However, if there is an iPad connected to that hub, it will consume 2100mA and the remaining 2900mA will be split for 6 ports, which can create an issue even for mobile phones, since it will be less than the required mA (480 instead of 500mA).

Bellow you can find a sample table for most popular devices with their sync and charge power requirement:

Device	mAmp	Device	mAmp
iPad Retina	2400	**Samsung S9/S9+**	2000
iPad 2	2100	**Samsung Note8**	2100
iPad Air and iPad Air 2	2100	**LG G4**	1800
iPad Mini 2 and 3	2100	**Google Pixel 2**	2000
iPad Mini	1000	**Huawei Mate 9**	2000
iPhone 6/7 and iPhone 6/7 Plus	1000	**Motorola Nexus 6**	2000
iPhone X and iPhone XS	2100	**Xiaomi Mi 5**	1000
iPhone 5s	500	**Lenovo K8**	1000
iPhone 8 and iPhone 8 Plus	2000		

The recommendation is to plan and calculate power requirements in advance, in order to avoid devices disconnections due to power issues.
Also, the recommendation is to use power USB hubs that comply with the BC 1.2 standard.
Below you can find the recommended models of USB hubs:

Pluggable USB 3.0 7-PORT CHARGING HUB WITH 60W POWER ADAPTER:
https://plugable.com/products/usb3-hub7bc/

Cambrionix PowerPad15:
https://cambrionix.com/products/powerpad15c-managed-usb-charger/

Device/s hosting

Since the mobile devices are constantly connected to a power source, you should perform some actions in order to reduce amount of heat and impact that this configuration creates.
Follow these recommendations:

- Place the devices in a non-flammable, well ventilated enclosure
- Provide extra ventilation for the enclosure
- Have enough space between the devices, so they will not overheat
- Reduce device screen brightness to a minimum (to avoid extra heat and screen damage)
- Use only original USB cables that comes with the phone

There are solutions in the market that are meeting those requirements.
See example: https://www.tripplite.com/16-port-usb-tablet-charging-station-white~CS16USBW

Devices bean for rack-mounted installation

Extra-fan panel for rack-mounted instantiation

1U 16 ports USB power hub

16-Device USB Charging Station Cabinet

See additional best practices related to the devices hosting: https://admhelp.microfocus.com/mobilecenter/en/latest/Content/Configuring_and_connecting_devices.htm#mt-item-3

Device configuration

To help with the device configurations, see the checklist for connecting a device to Mobile Center:

Action	Done	Remarks
No passcode configured on the device	☐	
No Google Play Account/Apple Store Account configured on the device	☐	
Device connected to the Wi-Fi	☐	
Device screen brightness to minimum	☐	
Device wallpaper set to monochrome, static	☐	
Android		
Disable Lock device option	☐	
Disable Wallpaper option	☐	
Enable Developer option (Go to Settings →About Device → Click 7 times on Build number)	☐	
Enable Stay Awake option under developer option	☐	
Enable USB Debugging option under Developer option	☐	
On Samsung device that run Android 8.0 please make sure to add MC agent to unmonitored application under Battery saver menu	☐	
Check the Chrome version (Launch Chrome browser →Help → Version info)	☐	
Disable auto-update and patches install	☐	
iOS (Apple)		
Copy the UUID of the device (required for Agents resign)	☐	
Disable the Lock device option	☐	
If the device is under iOS 11.2.5 configure the Auto-Lock to Never	☐	

If the device is 11.2.5 and above configure the Auto-lock to 30 Seconds	☐	
Under Setting →Safari → Advanced enable JavaScript and Web Inspector option	☐	
Enable UI Automation option (after first connection to MC the option will appear in the Settings)	☐	
Disable the iOS auto update (Go to settings →General →iPhone Storage)	☐	

To avoid automatic iOS upgrades on the mobile devices, follow these guidelines:

- Tap Settings.
- Tap iTunes & App Store.
- In the section headed Automatic Downloads, set the slider next to Updates to **Off**.

To remove already downloaded updates:

- Open the Settings app.
- Tap General.
- Tap iPhone/iPad Storage.
- If you scroll down a little, you'll see a list of apps and the amount of storage they each take up. Somewhere in there you'll find the iOS update.
- Tap the update to see more details, and then select **Delete Update**.
- Tap Delete Update to confirm.

It is also possible to block the automatic updates by blocking the following domains on the WiFi router: **appldnld.apple.com** and **mesu.apple.com**

For Android devices:

- Settings > System > About device > Software update. Deselect auto update

Additional items to consider:

- SIM card error message: this system alert message can prevent plug-n-play operation for the device. Solution: Install a fake sim card or use MC Agent solution to overcome it (documented in Mobile Center Help)
- Automatic dismissal of system dialogs (MC Agent setting documented in MC Help)
- Automatic prevention of device lock (MC Agent setting documented in MC Help)

MAINTENANCE OPERATIONS

Mobile Lab Inspection

Due to nature of the setup, there is a need to perform periodical physical inspection in the mobile lab. The purpose of that inspection is to review the current setup and assure that there is no damage that can impact the system.
Example of an inspection checklist:

Action	Done	Remarks
Check that all devices are connected to Wi-Fi	☐	
Browse from several devices to check if the Wi-Fi network available	☐	
Check each physical device for a swollen battery	☐	When lithium-ion batteries are overheated, over-charged, or simply reach an old age, it is possible for the inner cells of the battery to emit a flammable electrolyte mixture and cause a swollen battery.
Check that all devices are charging and that the battery level is 100%	☐	
Check that device brightness is set to minimum	☐	
Check that devices are not locked	☐	
Check that no unwanted OS installs were downloaded to the devices (OS upgrades or patches)	☐	

Database Maintenance

PostgreSQL, like any database software, requires that certain tasks be performed regularly to achieve optimum performance.
The following procedures are the most common:

- The creation of backup copies of the data on a regular schedule.
- Periodic "vacuuming" of the database.
- Audit Log file management (by default, PostgreSQL is installed with Audit log activated. To deactivate the feature: update file PostgreSQL\9.6\data\postgresql.auto.conf to setting logging_collector = 'off'

For more information, see https://www.postgresql.org/docs/9.6/static/maintenance.html.

Logs and TMP cleanup

Despite the fact that the logs of MC are remove older data, there are still some conditions that will cause certain log files to grow significantly. For example, the application packager log, MC audit.log, DB audit log, and so forth.
There is a need to monitor the size of those logs and perform cleanup actions from time to time.

MONITORING

Like any other production system, Mobile Center deployment requires monitoring for performance and availability.
The following types of monitoring should be applied:

- Hardware: memory, CPU, disk space, network consumption
- Services: process/service availability
- Network availability: URL monitoring
- Device availability
- Connector availability
- Database performance (PostgreSQL: https://bucardo.org/check_postgres/)
- Log files monitoring for exceptions and errors

Mobile Center provides various ways to achieve that:

- REST API (http://mobilecenter.microfocus.com/api/): any action related to MC can be executed via REST API. It can be used in scripting for monitoring purposes.
- Embedded statistics reporting engine: The MC Server aggregates statistics from the connector and exposes them via the statsD daemon or Prometheus reporter.

Mobile Center Log files are located in the /log folder.

UPGRADE PROCESS

Due to the critical business value of the system, it is important that the upgrade process be rolled out in very organized and robust way. Follow these best practices:

- Never upgrade in place: use two environments, your current system and another, a new installation, running in parallel. Follow the procedure as how to migrate an MC servert: https://admhelp.microfocus.com/mobilecenter/en/latest/Content/migrate_server.htm

- Make backups: not just before an upgrade, but regularly. MC doesn't store transactional data in the DB, but it is still good practice to keep your data safe.
- Compatibility check: allow your end users to re-run their tests and actions with a new system to assure compatibility of their assets with the new version before going live.
- Leverage tools provided by the vendor: do not try to adopt/modify the system manually. Example: migration tool for mobile applications
- Plan the migration and execute: plan your actions before, during and after the upgrade.

The following flow can be suggested for typical upgrade process (Staging env):

- Backup the current MC Production DB and restore it on a staging env.
- Backup the current MC Production settings (LDAP config, extended integrations, etc.). MC stores all this information in the DB, but you may still want to take few screenshots for your convenience.
- Upgrade the iOS Packaging service which will be used for Agents distribution post an upgrade.
- Perform an MC Server upgrade. Note: Consider not to upgrade the applications during the upgrade itself, since it adds a significant amount of time (1-2 min per each app version). You can always run the standalone application upgrader tool afterwards.
- Verify the system data integrity (users, tenants, apps)
- Perform upgrade on the Connectors machines and verify that they are working
- Redistribute new Agents to the Connectors (Connectors page)
- Perform a devices reconnect (Connector page and verify devices availability
- Perform end2end verification of the device usage flow
- Run Mobile Apps upgrade (this can take some time, depends on amount of the apps)

Note: If no iOS packaging service is used, perform a manual re-sign of new iOS agents and distribute it to the connectors.

In case of any failure in the upgrade process, please verify the installation/upgrade log.

PACKAGING SERVICES

Mobile Center works with both packaged and non-packaged mobile apps. Packaging is an instrumentation method that injects the MC intercept library into the application bundle and also re-signs the app with proper credentials. The advantage of using packaged apps is to provide better object recognition for record/replay as well as additional sensors simulations (photo, fingerprint, etc.).
After you upload an app to Mobile Center, the server automatically attempts to package the app. This provides users with the option of selecting either a packaged app or the original version when running a test. To enable this functionality of automatic app packaging and signing by MC, the administrator needs to set up the packaging and signing services.

The packaging service is also used during the upgrade process, when the existing app being upgraded with latest version of the instrumentation library.
General information about packaging services, including manual procedure for packaging the apps is explained here:
https://admhelp.microfocus.com/mobilecenter/en/latest/Content/_lp_prepare_your_app_for_upload.htm

Android packaging

By default, the Android packaging service is installed together with Mobile Center Server. It does not require any special configuration but it can impact overall performance of MC Server machine since the packaging service is a Java process which runs on the server.

iOS packaging

The packaging procedure for iOS apps is slightly different. First, due to Apple constraints, the packaging procedure for IOS apps can only be done on OSX, so there is a requirement for Mac machine and XCode environment. Secondly, Apple Developer Certificate and Provisioning Profile must be used. In an iOS application's bundle, you'll find the Entitlements.plist, which is a list of capabilities that an application allows. When signing your application using a certificate intended for distribution, the signing utility will not allow an entitlement with get-task-allow set to YES. This is because get-task-allow is what allows the MC library to connect to a process, and Apple does not want that on apps meant for distribution. Therefore, a Development Certificate and Provisioning profile for Development must be used.

Those parameters should be configured in `/conf/packager.properties` file on the Mac machine where the packager service is installed. For installation instructions, see https://admhelp.microfocus.com/mobilecenter/en/latest/Content/AutomaticPackagingService.htm .

Once the MC iOS packaging service is installed, you can also access it via a browser using the URL: `http://<MAC_MACHINE_ADDR>:<PORT>/instrumentation/index.html` (use Chrome in order to allow the download functionality of the packaged app).