UFT Mobile Administrator

Best Practices

MICRO FOCUS

Contents

Installation and configuration **3**

General deployment considerations 3
Deployment scenarios 4
Hardware requirements 5
Network requirements 6
Network latency *6*
UFT Mobile and SSL ..... *7~~6~~*
UFT Mobile ports *7*
Client tools and UFT Mobile server connectivity *7*
Connector scalability 7
USB hubs and device power consumption ..... *8~~7~~*
Device/s hosting *9*
Device configuration *10*

Maintenance operations **12**

Mobile lab inspection 12
Database maintenance 13
Logs and TMP cleanup 13

Upgrade process **14**

Packaging services 15
Android packaging *15*
IOS packaging *15*

Installation and configuration

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

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

General deployment considerations

UFT Mobile supports a distributed architecture in which different test clients can interact with the same UFT Mobile server instance.

UFT Mobile deployment has a number of components:

Component	Function

UFT Mobile Server	This is a single web server that can be installed on a physical or virtual environment. It serves to:
	• Mediate between the testing-tool client calls to mobile devices, and provide a user interface within the testing tool for recording and running tests on real mobile devices.
	• Accept apps for testing and manages app versions.
	• Provide 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 various 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.
	Note: When you install the UFT Mobile server, you have the option to install an embedded connector if you want to connect devices directly to the UFT Mobile server instance.

PostgreSQL database	You can choose either to connect UFT Mobile to an existing external PostgreSQL database, or use the database that is embedded in the UFT Mobile Server installation (physical or virtual). You specify this option during installation. For details, see UFT Mobile - Windows Installation or UFT Mobile - Linux Installation.

Component	Function
Connector	The connector is designed as a lightweight piece of software for connecting devices to UFT Mobile, and can 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 a virtual environment; however, it must maintain USB connectivity to the devices (USB pass-through for mobile devices).
High Availability	You can configure high availability in an active-passive configuration using multiple servers. In this mode there is one active UFT Mobile server, to which the load balancer routes all the requests, and another passive UFT Mobile server ready to take over in case the active server fails. For details on this configuration, see High Availability support in UFT Mobile (on-premises).
File Storage System	Applications are no longer stored in the database but are saved to the file system. When installing or upgrading, you can select a destination folder for storing applications uploaded to UFT Mobile. You can also control the number of uploads per application and choose to automatically delete old uploads of an application. This makes it easier for the UFT Mobile administrator to manage the number of application uploads that need to be maintained, and reduces load on the file storage system. For details on this feature, see the section File storage configuration section under Install UFT Mobile on a Windows machine and the section Limit application uploads under General settings.

Deployment scenarios

The decision point for UFT Mobile deployment scenario varies according to customer requirements.

Scenario	Description	Advantages
All-in-one	Single box deployment for UFT Mobile server, database, and embedded connector.	Simplicity. Ideal for proof of concept and local installations.
3-Tier deployment	Separate web and data layers by installing UFT Mobile server and databases on different locations.	Scalability of web and database layers. Supports local IT best practices for web and database management.

For the deployment of connectors/devices, the following scenarios can be considered:

Scenario	Description	Advantages
Central device hub	A central lab of devices connected to the connector on the UFT Mobile 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 hardware requirements for UFT Mobile is available at https://admhelp.microfocus.com/mobilecenter/en/latest/Content/Before_starting_installation.htm

When planning UFT Mobile hardware resources, take the following parameters into consideration:

Component	Memory	CPU	Disk Space
UFT Mobile Server	UFT Mobile Server is a Java application. Therefore, it uses a predefined amount of host memory. The amount of consumed memory is impacted by the number of simulation sessions (user sessions). The minimal memory requirement is 4 GB. We recommend 8 GB for medium deployment (<30 devices), and 16 GB for large deployment (>30 devices).	UFT Mobile Server CPU consumption is dependent on the number of requests that are processed. The minimal requirement is an x64 processor, 2.2 GHz	Disk space usage on the UFT Mobile Server depends on the number of factors such as logs generated, packaged applications, and processes. We recommend at least 20 GB: 15 GB for general installation and 5 GB for the temporary folder. Please note that in versions 3.5 and above, you can specify a temporary folder different than TMP/TEMP An additional 1 GB of free disk space is required on the system disk.

Component	Memory	CPU	Disk Space
PostgreSQL DB	PostgreSQL memory consumption is impacted by SQL queries that it is required to execute. Minimal requirement for memory is 2 GB. We strongly recommend at least 8 GB for medium deployment (<30 devices), and 16 GB for large deployment (>30 devices).	PostgreSQL is process based. The minimal requirement is a dual-core CPU, 2.2 GHz.	Disk space usage on PostgreSQL depends on the data size. UFT Mobile 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	UFT Mobile Connector is a Java application. Hence, it uses a predefined amount of host memory. The amount of consumed memory is impacted by the number of simulated sessions (user sessions). The minimal requirement is 2 GB. We recommended at least 8 GB for standard deployments (8-10 devices per connector), and 16 GB for large deployments (12-25 devices).	The guidelines for UFT Mobile Server are the same for the Connector Java application. Remote access to the device increases the 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. The rule of thumb is to allocate one-half of the CPU Core for each remote device session.	The disk space usage on the UFT Mobile Connector depends on various factors, such as the number of logs generated, and the number of application files cached on the connector. We recommend at least 10 GB.

Network requirements

UFT Mobile provides straightforward network requirements. You can find complete information about UFT Mobile architecture at https://admhelp.microfocus.com/mobilecenter/en/latest/Content/Architecture.htm

Network latency

UFT Mobile is designed for resiliency over the network (WAN), by using REST API communication over the HTTP/S protocol. However, there is also a communication channel that leverages the WebSocket protocol. Communication through this protocol may present some limitations that need to be considered.

In general, if network latency is less than 100 ms, communication issues are unlikely when UFT Mobile and connectors are using the public Internet, MPLS, VPN, or any other method. A latency greater than 200 ms will introduce connectivity challenges.

To work on a device in remote view, we recommend network bandwidth of 1 Mbps or higher.

UFT Mobile and SSL

By default, UFT Mobile uses an SSL configuration to communicate between a server and connectors. This is achieved by generating a self-signed SSL certificate during the installation. For production usage, we strongly recommend using CA certificates (certificate issued by Certification Authority as opposed to self-signed), which will remove security warnings in browsers as well as streamline connectivity of testing tools. We also recommend using 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

Using SSL is also beneficial from a networking perspective, as it eliminates any internal security blockages by IPS or other security gateways.

UFT Mobile ports

UFT Mobile Server (Web front end) utilizes a single port. The port is configured during the installation of UFT Mobile Server. The UFT Mobile connector also utilizes a single port for connectivity with the UFT Mobile server and the end-user (client). Internally, the UFT Mobile connector utilizes a reverse-proxy (Nginx) to route the requests to relevant mobile devices. Therefore, from the networking perspective, a single port should be accessible (ingress) for the UFT Mobile Server and Connector.

Regarding protocols used, there is a requirement for HTTP/S and WebSocket/WebSocket Secure (WS/S) protocols.

Client tools and UFT Mobile server connectivity

Common client tools are UFT One, LoadRunner, Sprinter, BPM, UFT Developer, and Appium scripts.

Testing-tool clients connect to the UFT Mobile 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 can handle a significant number of mobile devices. However, the maximum number of devices per single connector is defined by several parameters, such as operating system, motherboard hardware, USB ports, and their versions.

For example, Windows 7 has limitations relating to USB3.0 ports (supported natively in Windows 8). Therefore, we recommend the following:

- Avoid using Windows 7 for a UFT Mobile Connector machine (not part of supported configuration).
- Connect up to 20 mobile devices per single connector (using USB hubs; see the following section).
- For iOS-based deployments, consider using the UFT Mobile Connector for OSX.
- For Android-based deployment, consider using the UFT Mobile Connector for Linux or Windows (Android device drivers need to be installed separately).
- Ensure the OS is not configured for hibernation or sleep, to keep devices stable in the OS.

USB hubs and device power consumption

When a device is used with UFT Mobile, there is a need for synchronization and charging. The device is connected via a USB cable, which provides constant charging and communication (UFT Mobile Connector to Agent).

As the number of USB ports is usually limited, use a USB self-powered hub to support the required scalability. The hub is powered by an external power supply and can therefore provide full power to every port.

Charging requirements for mobile devices vary from 500 to 3,000 mA (from Android and iOS phones to tablets and iPads). We strongly recommend that you ensure the power hub can deliver the required power to all USB ports.

Consider, for example: A powered 7-port USB hub of 60 W has specs of 12V and 5A (12x5=60). A smart hub dynamically splits the 5A among 7 ports, giving each port ~714 mA, which is sufficient for small/older mobile phones. However, if an iPad is connected to that hub, it will consume 2100 mA, leaving the remaining 2900 mA to be split among 6 ports (~480 mA each); this might be an issue even for mobile phones, since the power allotment is less than the required 500 mA.

The following table lists the most popular devices and their power requirement for sync and charge.

iOS Devices	mA	Android Devices	mA
iPad Pro 12.9 inch (4th generation)	3000	Samsung S9/S9+	2000
iPad Pro 12.9 inch (3rd generation)	3000	Samsung Note8	2100
iPad Pro 11-inch (2nd generation)	3000	LG G4	1800
iPad Pro 11-inch	3000	Google Pixel 2	2000
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 5s	500	Lenovo K8	1000
iPhone 6/7 and iPhone 6/7 Plus	1000	Motorola Nexus 6	2000
iPhone X and iPhone XS	1000	Xiaomi Mi 5	1000
iPhone 8 and iPhone 8 Plus	1000		
iPhone XS Max	1000		
iPhone XR	1000		
iPhone 11	2000		
iPhone 11 Pro	2000		
iPhone 11 Pro Max	2000		

We recommend that you plan and calculate power requirements in advance to avoid device disconnections due to power issues. In addition, we recommend that you use powered USB hubs that comply with the BC 1.2 standard. The following products are recommended:

Pluggable USB 3.0 7-Port Charging Hub with 60W Power Adapter

https://plugable.com/products/usb3-hub7bc/

Note: This USB hub supports only 900mA per port. We therefore recommend connecting no more than 4 devices to it.

16-Port USB Charging Station with Syncing, 230V, 5V 40A (200W) USB Charger Output, 2U Rack-Mount

https://www.tripplite.com/16-port-usb-charging-station-syncing-230v-5v-80a-400w-2u-rack-mount~U280016RMINT

Cambrionix PowerPad 15

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

Device hosting

The mobile devices are constantly connected to a power source, therefore we recommend the following actions to reduce the amount of heat and impact of this configuration:

- Place the devices in a non-flammable, well-ventilated enclosure
- Provide extra ventilation for the enclosure
- Maintain enough space between the devices to prevent them from overheating
- Reduce device screen brightness to a minimum to avoid excessive heat and screen damage
- Use only the original USB cables supplied with the phones

A number of solutions are available to help you meets these requirements. See, for example:
https://www.tripplite.com/16-port-usb-tablet-charging-station-white~CS16USBW

Devices beam for rack-mounted installation

Extra-fan panel for rack-mounted instantiation

1U 16 ports USB power hub

16-device USB charging station cabinet

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

Device configuration

To help with device configurations, refer to the checklist for connecting a device to UFT Mobile:

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	☐	

Action	Done	Remarks
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 options	☐	
Enable USB Debugging option under Developer options	☐	
On Samsung device that run on Android 8.0, make sure to add the UFT Mobile Agent to unmonitored applications 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 running on a iOS version earlier than 11.2.5 configure the Auto-Lock to Never	☐	
If the device runs on iOS 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 UFT Mobile 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:

1. Tap **Settings**.
2. Tap **iTunes & App Store**.
3. In the section **Automatic Downloads**, turn off the **Updates** slider.

To remove previously downloaded iOS updates:

1. Open the **Settings** app.

2. Tap **General**.
3. Tap **iPhone/iPad Storage**.
4. Scroll down slightly until you see a list of apps and the amount of storage they use. Look for the iOS update.
5. Tap the update to see more details, and then select **Delete Update**.
6. Tap **Delete Update** to confirm.

You can also block iOS automatic updates by blocking the following domains on the Wi-Fi router:

- appldnld.apple.com
- mesu.apple.com

To avoid automatic upgrades on the 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-and-play operation for the device. Solution: Install a fake sim card or use the UFT Mobile Agent solution to resolve (see UFT Mobile Help)
- Automatic dismissal of system dialogs (UFT Mobile Agent setting, see UFT Mobile Help)
- Automatic prevention of device lock (UFT Mobile Agent setting, see UFT Mobile Help)

Maintenance operations

Mobile lab inspection

Due to nature of the setup, you must periodically perform a physical inspection of the mobile lab. The purpose of that inspection is to review the current setup and ensure there is no damage that can impact the system.

The following table is an 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, the inner cells of the battery may emit a flammable electrolyte mixture, causing the battery to swell.

Action	Done	Remarks
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 certain tasks to be performed regularly to achieve optimum performance.

The following procedures are the most common:

- 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\<version>\data\postgresql.auto.conf by setting logging_collector = 'off'

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

Logs and TMP cleanup

Even though the UFT Mobile logs remove older data, some conditions cause certain log files to grow significantly. For example, the application packager log, UFT Mobile audit.log, and database audit log.

You need to monitor the size of these logs and periodically perform cleanups.

Monitoring

Like any other production system, UFT Mobile deployment requires monitoring for performance and availability.

The following types of monitoring are necessary:

- 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/)
- Monitoring log files for exceptions and errors

UFT Mobile provides various methods for effective monitoring:

- REST API (https://admhelp.microfocus.com/documents/mobilecenter/api/3.4/). Any action related to UFT Mobile can be executed via REST API. REST API calls can be used in a script for monitoring purposes.
- Embedded statistics reporting engine. The UFT Mobile Server aggregates statistics from the connector and exposes them via Prometheus reporter.

UFT Mobile Log files are stored in the **/log** folder.

Upgrade process

Because of the system's vital business value, the upgrade process must be rolled out in very organized and robust way.

Be sure to follow these best practices:

- **Never upgrade in place**. Use two environment – your current system and another, new installation running in parallel. Follow the procedure for migrating an UFT Mobile server: https://admhelp.microfocus.com/mobilecenter/en/latest/Content/migrate_server.htm
- **Backup**. Backup regularly, not only before an upgrade. UFT Mobile does not store transactional data in the database, but it is still good practice to keep your data safe.
- **Compatibility check**. Allow end users to rerun 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. Use a migration tool for mobile applications, for example.
- **Plan the migration and execution**. Plan your actions before, during, and after the upgrade.

The following flow is suggested for a typical upgrade process (staging environment):

- Backup the current UFT Mobile production database and restore it on a staging environment.
- Backup the current UFT Mobile production settings (such as LDAP configuration and extended integrations). UFT Mobile stores all this information in the database, but you might want to capture a few screenshots for reference.
- Upgrade the iOS Packaging service that will be used for Agents distribution after an upgrade.
- Perform an UFT Mobile Server upgrade.

  Note: Consider not upgrading the applications during the upgrade itself, because 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 an upgrade on the connector machines and verify they are working.
- Redistribute new Agents to the Connectors (Connectors page).

- Perform a device reconnection (Connector page) and verify devices availability.
- Perform end-to-end verification of the device usage flow.
- Run Mobile Apps upgrade (this can take some time, depending o the number of apps).

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

If any part of the upgrade process fails, check the installation/upgrade log.

Packaging services

UFT Mobile works with both packaged and non-packaged mobile apps. Packaging is an instrumentation method that injects the UFT Mobile intercept library into the application bundle and 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 (such as photo or fingerprint).

After you upload an app to UFT Mobile, the server automatically attempts to package the app. This gives users the option of selecting either a packaged app, or the original version when running a test. To enable the functionality of automatic app packaging and signing by UFT Mobile, the administrator needs to set up the packaging and signing services.

The packaging service is also used during the upgrade process, when the current app is upgraded with latest version of the instrumentation library.

For general information about packaging services, including manual procedure for packaging the apps, see 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 UFT Mobile Server. It does not require any special configuration, but it can impact overall performance of UFT Mobile Server machine because the packaging service is a Java process that 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 be done only on OSX, so there is a requirement for a Mac machine and an XCode environment. Second, Apple Developer Certificate and Provisioning Profile must be used. In an iOS application bundle, you will 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**. Because get-task-allow enables the UFT Mobile library to connect to a process, Apple does not want this entitlement enabled on apps intended for distribution. Therefore, a Development Certificate and Provisioning profile for Development must be used.

The parameters need to 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 UFT Mobile 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 to allow the download functionality of the packaged app.)