opentext™

Digital Lab

Administrator Best Practices

Contents

1 Installation and configuration

Digital Lab 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.

1.1 General deployment considerations

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

Digital Lab deployment has several components:

Component	Function

Digital Lab 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 manage 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 Digital Lab server, you have the option to install an embedded connector if you want to connect devices directly to the Digital Lab server instance.

PostgreSQL database	You can choose either to connect Digital Lab to an existing external PostgreSQL database, or use the database that is embedded in the Digital Lab Server installation. You specify this option during installation. For details, see Digital Lab - Windows Installation or Digital Lab - Linux Installation.
Connector	The connector is designed as a lightweight piece of software for connecting devices to Digital Lab 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.

Component	Function
	The connector manages the physical USB connection to the device, and the logical state machine on top of it. The connector can be installed in 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 Digital Lab server, to which the load balancer routes all the requests, and another passive Digital Lab server ready to take over in case the active server fails. For details on this configuration, see High Availability support in Digital Lab (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 Digital Lab. 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 Digital Lab administrator to manage the number of application uploads that need to be maintained and reduces the load on the file storage system. For details on this feature, see the section File storage configuration section under Install Digital Lab on a Windows machine and the section Limit application uploads under General settings.

1.2 Deployment scenarios

The decision point for Digital Lab deployment scenario varies according to customer requirements.

Scenario	Description	Advantages
All-in-one	Single box deployment for Digital Lab server, database, and embedded connector.	Simplicity. Ideal for proof of concept and local installations.
3-Tier deployment	Separate web and data layers by installing Digital Lab 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 Digital Lab server machine.	Efficiency. Avoids duplication of tasks for setting up and managing devices.

Scenario	Description	Advantages
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.

1.3 Hardware requirements

The full list of hardware requirements for Digital Lab is available in the Support Matrix online documentation.

When planning Digital Lab hardware resources, consider the following parameters:

Component	Memory	CPU	Disk Space
Digital Lab Server	Digital Lab 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). * Based on your machine memory we recommend you increase the maximum heap size for the Java virtual machine (JVM) when installing, upgrading, or modifying the Digital Lab server for example: if the machine had 8 GB you can increase the Java heap size to 4GB.	Digital Lab 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 Digital Lab Server depends on several 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. The 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. On Windows, PostgreSQL is installed on the C: drive, so disk space must be allocated there.
Connector	Digital Lab 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 recommend at least 8 GB for standard deployments (8-10 devices per connector), and 16 GB for large deployments (12-25 devices).	The guidelines for Digital Lab 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 Digital Lab 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.

1.4 Network requirements

Digital Lab provides straightforward network requirements. You can find complete information about Digital Lab architecture in the Digital Lab Architecture topic.

1.4.1 Network latency

Digital Lab 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 Digital Lab 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 a network bandwidth of 1 Mbps or higher.

1.4.2 Digital Lab and SSL

By default, Digital Lab 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 the topic Working with SSL and certificates.

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

1.4.3 Digital Lab ports

Digital Lab Server (Web front end) utilizes a single port. The port is configured during the installation of Digital Lab Server. The Digital Lab connector also utilizes a single port for connectivity with the Digital Lab server and the end-user (client). Internally, the Digital Lab 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 Digital Lab Server and Connector.

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

1.4.4 Client tools and Digital Lab server connectivity

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

Testing-tool clients connect to the Digital Lab 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/WSS).
- The remote screen viewer client sent over WebSocket (WS/WSS)

1.5 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 USB 3.0 ports (supported natively in Windows 8). Therefore, we recommend the following:

- Avoid using Windows 7 for a Digital Lab Connector machine (not part of the supported configuration).
- Connect a maximum of 25 mobile devices per single connector (using USB hubs; see the following section).
- For iOS-based deployments, consider using the Digital Lab Connector for OSX.
- For Android-based deployment, consider using the Digital Lab 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.

1.5.1 USB hubs and device power consumption

When a device is used with Digital Lab, there is a need for synchronization and charging. The device is connected via a USB cable, which provides constant charging and communication (Digital Lab 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 5,000 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

iOS Devices	mA	Android Devices	mA
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	Samsung S20/S20+	4000
iPhone XS Max	1000	Samsung S21 Ultra	5000
iPhone XR	1000	Samsung S21/S21+	4000/4800
iPhone 11	2000	Google Pixel 4a	3140
iPhone 11 Pro	2000	Google Pixel 5	2800
iPhone 11 Pro Max	2000	Motorola One 5G	5000
iPhone 12	2815	Samsung S22 Ultra	5000
iPhone 12 Pro	2815	Google Pixel 6 Pro	5000
iPhone 12 Pro Max	3687	Samsung Galaxy Z Flip 3	3300
iPhone 13	3227	Oppo Find X5 Pro	5000
iPhone 13 Pro	3095	Samsung S22+	4500
iPhone 13 Pro Max	4352	OnePlus Nord 2	4500
iPad mini 6	8827	OnePlus 10 Pro	5000
		Xiaomi Redmi Note 10 Pro	5020
		Xiaomi Mi 11	4600

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. Here are some examples of products recommended:

16-Port USB 2.0 hub 200W multiple USB port hub - USB charging splitter 5V 40A

https://brovss.shop/products/usb-hub-a173

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

SuperSync15 – Cambrionix Multideck

https://www.cambrionix.com/products/supersync15-industrial-usb-3-hub-cambrionix-multideck

1.5.2 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 meet 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 the topic Connect devices to Digital Lab --> Physical device connectivity

1.5.3 Device configuration

To help with device configurations, refer to the checklist for connecting a device to Digital Lab:

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	☐	
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 and later, make sure to add the Digital Lab Agent to unmonitored applications under Battery saver menu	☐	
Disable auto-update and patches install	☐	
iOS (Apple)		
Copy the UDID of the device (required for Agents resign)	☐	
Disable the Lock device option	☐	
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 Digital Lab the option will appear in the Settings)	☐	
Disable the iOS auto update (Go to settings → General → Software Update)	☐	

To avoid automatic upgrades on iOS devices:

1. Tap **Settings**.

2. Tap **General**.
3. In the section **Software update**, turn off the **Automatic Updates** option.

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 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 Digital Lab Agent solution to resolve (see Digital Lab Help)
- Automatic dismissal of system dialogs (Digital Lab Agent setting, see Digital Lab Help)
- Automatic prevention of device lock (Digital Lab Agent setting, see Digital Lab Help)

2 Maintenance operations

2.1 Mobile lab inspection

Due to the nature of the setup, you must periodically perform a physical inspection of the mobile lab. The purpose of this inspection is to review the current setup and ensure no damage 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	☐	
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.
Check that all devices are charging and that the battery level is 100%	☐	
Check that device brightness is set to minimum	☐	
Check that the devices are not locked	☐	

2.2 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

For more information, see https://www.postgresql.org/docs/11/static/maintenance.html.

2.3 Logs and TMP cleanup

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

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

3 Monitoring

Like any other production system, Digital Lab 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

Digital Lab provides various methods for effective monitoring:

REST API - Any action related to Digital Lab can be executed via REST API. REST API calls can be used in a script for monitoring purposes.

- Embedded statistics reporting engine. The Digital Lab Server aggregates statistics from the connector and exposes them via Prometheus reporter.

Digital Lab Log files are stored in the **/log** folder.

4 Upgrade process

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

Be sure to follow these best practices:

- **Never upgrade in place**. Use two environments – your current system and another, new installation running in parallel. Follow the procedure for migrating the Digital Lab server.
- **Backup**. Backup regularly, not only before an upgrade. Digital Lab 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 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.

For full details, please consult the best practices for upgrades.

4.1 Packaging services

Digital Lab works with both packaged and non-packaged mobile apps. Packaging is an instrumentation method that injects the Digital Lab 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 Digital Lab, 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 Digital Lab, 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 the latest version of the instrumentation library.

For general information about packaging services, including the manual procedure for packaging the apps, please visit the online help.

4.1.1 Android packaging

By default, the Android packaging service is installed together with Digital Lab Server. It does not require any special configuration, but it can impact the overall performance of the Digital Lab Server machine because the packaging service is a Java process that runs on the server.

4.1.2 IOS packaging

The packaging procedure for iOS apps is slightly different.

iOS applications and agents can be signed/packaged using the embedded packaging service or a remote packaging service. The remote packaging is required for installations with more than 100 mobile iOS devices or if there is a need to use more than one developer account for different iOS devices.

For further information, please consult the online help documentation.

About OpenText

OpenText enables the digital world, creating a better way for organizations to work with information, on-premises or in the cloud. For more information about OpenText (NASDAQ/TSX: OTEX), visit opentext.com.

Connect with us:

OpenText CEO Mark Barrenechea's blog

Twitter | LinkedIn