MobileIron Web@Work 2.5.0 for Android Guide

for MobileIron Core and MobileIron Cloud

October 27, 2020

Contents

New features summary

For the summary of new features introduced in previous releases, see MobileIron Web@Work Product Documentation for that release.

This release introduces the following new features and enhancements:

- Web@Work app features and enhancements
- Web@Work administrator features and enhancements

Web@Work app features and enhancements

- **Support for Android 11**: Web@Work 2.5.0 supports for Android 11. For additional information, see the Support and compatibility section.
- **Rebranding**: MobileIron has updated the Web@Work for Android icon and user interface color scheme.
- **Support for Android AppStation**: Web@Work now supports MobileIron AppStation as an additional Cloud client for a MAM-only deployment. A MAM-only deployment allows you to distribute and manage apps without having to manage the device itself.

Web@Work administrator features and enhancements

This release includes no new administrator feature and enhancement.

Download location

The current version is available for download here:

https://support.mobileiron.com/mi/android-browser/current/

Overview of Web@Work for Android

The following section provides an overview of the Web@Work app for Android devices:

About Web@Work

MobileIron Web@Work is a secure browser that allows enterprise users to securely access web content in their corporate intranet. Using Web@Work you can limit access to enterprise data to authorized users. When Web@Work is deployed in conjunction with AppTunnel, you secure the enterprise data in motion. The Web@Work app for Android is an AppConnect wrapped app.

Web@Work for Android is available in two flavors **Android AppConnect and Android AppStation**.

Web@Work Android AppConnect

Web@Work is available as an Android AppConnect app.

AppConnect is a MobileIron feature that containerizes apps to protect data on Android devices. Each AppConnect-wrapped app becomes a secure container whose data is encrypted, and protected from unauthorized access. Because each user has multiple business apps, each app container is also connected to other secure app containers. This connection allows the AppConnect apps to share data, such as documents. AppConnect apps are managed using policies configured in a MobileIron Enterprise Mobility Management (EMM) platform. The EMM platform is either MobileIron Core or MobileIron Cloud.

As an AppConnect app, all Web@Work data is secured. The app interacts with other apps according to the data loss prevention policies that you specify. You can also take advantage of AppConnect features such as app authorization and app configuration.

For information about AppConnect features and configuration beyond Web@Work for Android, see *MobileIron AppConnect and AppTunnel Guide*.

Web@Work for Android AppStation

Web@Work is available as an Android AppStation app for MobileIron Cloud only. MobileIron AppStation is specifically designed as the UEM client for a MAM-only deployment with MobileIron Cloud. In a MobileIron AppStation MAM-only deployment, only apps available through AppStation are managed. The following types of apps are supported:

- AppConnect apps (wrapped with Secure Apps Manager for AppStation)
- Non-AppConnect apps (in-house or from the Google Play Store)

Multi-factor authentication and authorization for device users

Device users can use Web@Work only if the following are true:

- The device and user are registered with MobileIron Core.
  Registering a device with Core *authenticates* the device user.
- The device is authorized to use Web@Work.
  Using the Admin Portal, you *authorize* a device to use Web@Work. You use Core's labeling mechanism to indicate which devices are authorized to use Web@Work.

NOTE: If the device is not authorized to use Web@Work, the device user cannot use it even for accessing public websites.

- The device is in compliance with the security policy applied to the device.
  Using the Admin Portal, you can set up security policies to block access to Web@Work if the device fails to meet conditions that you specify. When access is blocked, the device becomes unauthorized to use Web@Work. Also, all AppTunnel access is blocked, which blocks access to enterprise websites.
- Device users are logged in with their secure apps passcode.
  Web@Work is an AppConnect app, and therefore, you can optionally require the device user to enter a secure apps passcode to use it. The device user uses one secure apps passcode to access all AppConnect apps.
  When device users first launch Web@Work, they are prompted to create a secure apps passcode if they had not already created one to use some other AppConnect app. On subsequent launches of Web@Work, users are prompted to enter the secure apps passcode, unless they had recently entered it to use some other AppConnect app.

After device users have registered the device with MobileIron Core and, if required, entered their secure apps passcode, they require no further Web@Work setup.

NOTE: A device user cannot specify Web@Work as the default browser on the device. This prohibition ensures that the device user always has easy access to a browser for non-enterprise browsing, even if the device becomes unauthorized to use Web@Work.

Secure enterprise web content access using AppTunnel

Web@Work uses MobileIron's AppTunnel technology to securely access web content behind your enterprise's firewall. This technology allows you to:

- Set up Web@Work to access enterprise websites without requiring the device user to set up VPN.
- Support Single Sign On using Kerberos Constrained Delegation (KCD).
  Device users register Mobile@Work with MobileIron Core by entering their MobileIron credentials. They can then use Web@Work to access an enterprise app server without having to enter any further credentials. This support depends on the environment being set up to use KCD, and the necessary AppTunnel configuration.
- Limit enterprise access to Web@Work.
  Other apps, such as mobile email and calendar synchronization, are not impacted by Web@Work's enterprise access. Therefore, unlike when you use VPN for enterprise access, you do not have to retest the behavior of these existing apps.

- Limit the enterprise sites that a device user can access.
  You can specify accessible sites in the tunneling configuration. Specifically, as long as the device stays on the external network, internal sites that are not specified in the tunneling configuration remain inaccessible. Furthermore, you can vary the accessible sites according to device and user attributes, such as user membership in the enterprise directory.
- Terminate enterprise website access based on compliance policies.
  Using the security policy for a device, you can specify which non-compliance situations block AppTunnel access.
- Perform URL filtering to audit and enforce web use policies.
  If you direct all outgoing traffic through a filtering proxy, you can direct traffic that you tunnel through the proxy, too. For example, by setting up Web@Work to tunnel all requests to www.SomeExternalWebSite.com, you can set the URL rules in your filtering proxy to block access to that site.
- Benefit from split-tunneling.
  You can allow device users to access some public websites without tunneling, while enforcing tunneling for other external as well as enterprise websites. By setting up this split-tunneling, your device users can access public sites without incurring additional load on enterprise network infrastructure. In addition, split-tunneling allows users to access public websites without visibility to the enterprise. Regional privacy regulations sometimes require this for personally-owned devices.
- Secure tunneled web traffic using multi-factor authentication and authorization.
  To use Web@Work:
  - A device must be registered with MobileIron Core and authorized to use Web@Work.
  - You can optionally require a secure apps passcode to access Web@Work, in addition to the device passcode. Note that for Android devices running Secure Apps Manager 7.0 and earlier, a secure apps passcode is mandatory.

  Furthermore, establishing an AppTunnel requires a unique client-side certificate, ensuring that only managed and authorized devices can access enterprise websites. You can get certificates from a third-party certificate authority (CA) or from the CA built into MobileIron Core.

Enable MobileIron Access for Web@Work

Web@Work now supports MobileIron Access. MobileIron Access is a cloud service that secures access to enterprise content in business cloud services such as Office 365,G Suite, Salesforce, Box, and Dropbox. For information about MobileIron Access as a service and how to set up the service with MobileIron Core, see the *MobileIron Access Guide.*

Where to find Web@Work for Android

For the current download location, see the MobileIron Web@Work for Android Release Notes.

Support and compatibility for Web@Work for Android

For support and compatibility information, see the MobileIron Web@Work for Android Release Notes.

About Web@Work for Android configuration

Web@Work for Android, like all secure apps for Android, can only be distributed as an in-house app. When you distribute Web@Work, distributing the Secure Apps Manager is **required**.

You make Web@Work for Android available to device users as an **in-house app** in the App Catalog in the MobileIron Core Admin Portal (under **Apps > App Catalog > In-House**). The device user launches Mobile@Work for Android to discover and install Web@Work, where it will appear under Secure Apps within the Mobile@Work app.

What the users see in Web@Work for Android

When users launch We@Work for Android, they can access the following from the browsers screen:

- Back and forward arrows: Navigates through the browsed web pages. This option works if you have previously browsed some web page.
- Bookmarks: The bookmarks option, displays bookmarks if configured by admin on VSP.
- Opened tabs view
- Star icon: Bookmarks a web page by using this option. This option is enabled only of you have configured the bookmarks option.
- + icon: Opens a new tab.
- Settings: The following options are available:

Options	Functions
Refresh	The refresh option, refreshes the browser to show most updated version of the web page you are viewing.
Bookmarks	Displays the bookmarked the web pages that you want to use frequently.
History	The history option clears the browsing history.
Privacy	You can clear user sensitive such as: • Cache: Clears the cache. • History: Displays the browsing history.

Options	Functions
	• Cookies and Data: Clears the cookies and the browsing date and also closes all the open tabs. • Auto-fill form data: Saves data such as you name, address, contact information, and email address. When filling any form the auto-fill options saves time and automatically fills the required information. You can clear the auto-fill data. • Password: Saves passwords for different accounts. You can clear the stored passwords. • Location access • Camera access
About	The About options lists the following information about the Web@Work app: • Web@Work version • Chromium version • Build time • Build number • Licenses
Request desktop site	Enables desktop version of the site. This option is visible in settings only if the required key value pair are applied.
Logout	Clears all user sensitive data. This option is visible in settings only if the required key value pair are applied.

Web@Work features

Android versions of Web@Work support the same core functionality. However, some features of Web@Work may be specific to only one or the other operating system. Web@Work has the following features.

NOTE: Web@Work does not currently support video streaming.

Web@Work Feature	Platform Support	Description
Secure access to websites hosted on servers behind your firewall, without requiring the device user to use VPN	Android	Web@Work uses AppConnect and AppTunnel capabilities to provide this secure access. NOTE: You can use Web@Work without purchasing AppConnect for third-party or in-house apps and without purchasing AppTunnel. **Configuration:** See Enabling Web@Work on page 15
Support for Single Sign On using Kerberos Constrained Delegation (KCD)	Android	Device users register Mobile@Work with MobileIron Core by entering their MobileIron credentials. They can then use Web@Work to access an enterprise app server without having to enter any further credentials. This support depends on the environment being set up to use KCD, and the necessary AppTunnel configuration. See "Authentication using an identity certificate and Kerberos constrained delegation" in the MobileIron Sentry Guide.
Admin-specified bookmarks	Android	Web@Work supports bookmarks that you specify on the Admin Portal. **Configuration:** See Enabling Web@Work on page 15.
User-specified bookmarks	Android	Device users can add, name, and remove bookmarks that they create. Device users cannot delete or edit bookmarks that you specify in the Admin Portal.
Ability to provide different Web@Work-related settings to different devices and users	Android	By using MobileIron Core's labels, you can provide different Web@Work-related settings to different devices and users, depending on, for example, device attributes and user membership in the enterprise directory. See "Using labels to establish groups" in the MobileIron Core Device Management Guide or Connected Cloud Device Management Guide.
Web content presentation and interaction similar to Google Chrome.	Android	Web@Work for Android uses the Chromium engine.
User can open documents only in other secure apps.	Android	All Android secure apps have this behavior.
Prevent copy and paste between Web@Work and other apps.	Android	You can choose whether data can be copied and pasted between Web@Work and:

Web@Work Feature	Platform Support	Description
		• any other app • only other AppConnect apps • only within Web@Work itself This behavior is controlled by the AppConnect global policy for all AppConnect apps.
Encrypt downloaded documents and prevent sharing them outside of the secure container	Android	Screen capture can be disabled, as well. These behaviors protect documents from leaking to non-secure apps.
Delete downloaded documents based on device compliance status	Android	Downloaded documents are automatically wiped from the device in the following cases: • The device has been out of contact for the specified amount of time. • The device is retired.
Whitelist and blacklist URL filtering	Android	Using key-value pairs, you can configure a blacklist to block URLs, and a whitelist as an exception to blacklisted URLs.
Custom messages for blacklisted URLs	Android	Using a key-value pair, you can provide custom warning messages that appear when Web@Work navigates to a blacklisted URL.
Configurable Home Page	Android	Using a key-value pair, you can specify a URL to be loaded as the home page when the user opens a new tab. The home page URL will be loaded automatically only if there are no defined company bookmarks.
Configurable product name (user agent)	Android	Using a key-value pair, you can change the product’s user agent name that is visible to web servers.
Form-based auto-fill and HTTP authentication	Android	Using key-value pairs, you can configure Web@Work to enable saving data and/or passwords that users enter on forms, and on HTTP authentication forms.
User can clear cached data	Android	The user can clear cached data in the Privacy menu, including clearing the cache, history, cookies and data, auto-fill form data, passwords, location access, and camera access.

Web@Work Feature	**Platform Support**	**Description**
User can save preferences for location sharing	Android	Users can save their preference for sharing their location on a per-website basis, and clear their choice in the Privacy menu.
View web pages in desktop mode	Android	Using a key-value pair, you can enable users to view the desktop versions of web pages instead of the mobile versions.
Prevents opening multiple tabs	Android	Web@Work for Android has an option that prevents it from opening extra tabs, which is applicable only for AppConnect-enabled apps. If a third-party app (such as an AppConnect-enabled app) sends the EXTRA_APPLICATION_ID when it launches the Web@Work browser, the Web@Work app will create a tab associated with the third-party app. All links launched in the third-party app will be opened in an associated tab.

Configuring Web@Work for Android AppConnect

The following describe how to set up Web@Work for Android AppConnect:

Required configuration for Web@Work for Android deployment

The following configurations are required for Web@Work for Android deployment by the device users:

- MobileIron Enterprise Mobility Management (EMM) platform: MobileIron Core or MobileIron Cloud.
- (Optional) Sentry, with AppTunnel enabled (required if you want to secure connection using Sentry).
- An Android device that is registered with a MobileIron EMM.
- MobileIron client: Mobile@Work for MobileIron Core deployments; MobileIron Go for MobileIron Cloud deployments.

For supported versions see the *MobileIron Web@Work for Android Release Notes*.

Main configuration steps for Web@Work for Android AppConnect (Core)

A Web@Work license is required on MobileIron Core to enable support. Enabling this setting indicates that you have the required license to deploy Web@Work.

NOTE: Although Web@Work uses AppConnect capabilities, do not select Enable AppConnect For Third-party and In-house Apps in System Settings, unless you also purchased that license.

To enable Web@Work:

1. In the Admin Portal, go to **Settings > System Settings > Additional Products**.
2. Click **Licensed Products**.
3. Select **Enable Web@Work**.
4. Click **Save**.

Add Web@Work to the App Catalog

Add Web@Work to the MobileIron Core App Catalog. Adding to the App Catalog makes the app available in Apps@Work on the device. Users can download and install the app from Apps@Work. When you distribute Web@Work, distributing the compatible Secure Apps Manager is **required**.

Download Android secure apps

You can download Android secure apps that MobileIron provides using your single company login credentials. The Web@Work and the Secure Apps Manager are available for download here:

Secure app	Download location
Web@Work for Android	https://support.mobileiron.com/mi/android-browser/current/
Secure Apps Manager	https://support.mobileiron.com/mi/android-sam/current/

Upload Web@Work for Android to MobileIron Core

Web@Work for Android, like all secure apps for Android, can only be distributed as an in-house app. Use the MobileIron Core Admin Portal to upload the Web@Work for Android APK file, the Secure Apps Manager APK file, and other secure apps, to MobileIron Core just as you would any in-house app. Device users will download the apps from **Secure Apps** within the Mobile@Work for Android app.

To distribute the apps as in-house apps:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Add App**.
3. Click **In-House**.
4. Click **Browse** and navigate to and select the AppConnect app (.apk) you want to upload.
5. Click **Next**.
6. Follow the prompts to add the app.
   The default settings should work in most cases
7. Apply the app to a label.
   This makes the app available in Apps@Work for the devices in the label. Make sure that the Apps@Work web clip is also applied to the same labels.

For details about uploading in-house Android apps, see “Adding in-house apps for Android” in the Apps@Work Guide.

Set up a Standalone Sentry to support AppTunnel for Web@Work

Standalone Sentry configured for AppTunnel is required to secure the data (data-in-motion) that moves between secure apps and your internal corporate data sources. Setting up app tunneling is a two-step process.

1. Configuring an AppTunnel service in Standalone Sentry
2. Configuring AppTunnel rules in the Web@Work configuration.

Before you begin

Ensure that you have a Standalone Sentry that is set up for AppTunnel and the necessary device authentication is also configured. See “Configuring Standalone Sentry for app tunneling” in the MobileIron Sentry Guide.

NOTE: The Web@Work setting you configure will refer to the Certificate Enrollment or Certificates setting. Do not assign labels to the certificates settings. The certificates are distributed to the appropriate devices based on the Web@Work setting.

Configuring an AppTunnel service in Standalone Sentry

To configure an AppTunnel service for Web@Work on Standalone Sentry:

1. In MobileIron Core, go to **Services > Sentry**.
2. Edit the entry for the Standalone Sentry that supports AppTunnel.
3. In the AppTunnel Configuration section, under Services, click + to add a new service.
4. Use the following guidelines to configure an AppTunnel service for Web@Work.

Item	Description
Service Name	Use the dropdown to select <ANY> NOTE: <CIFS_ANY> is not relevant to Web@Work. Selecting <ANY> means that the Web@Work user can reach any of your internal servers. Typically, you do not want to restrict users' access. However, if you do want to restrict their access to internal servers, you can list the services here instead of selecting <ANY>. The service name is any unique identifier for the internal servers. For example, some possible service names are: • SharePoint • Human Resources The following characters are invalid: 'space' \ ; * ? < > " |. The Service Name is used in the Web@Work setting.
Server Auth	Select the authentication scheme for the Standalone Sentry to use to authenticate the user to the enterprise server: • Pass Through The Sentry passes through the authentication credentials, such as the user ID and password (basic, digest or NTLM authentication) to the enterprise server. • Kerberos The Sentry uses Kerberos Constrained Delegation (KCD). KCD supports Single Sign On (SSO). SSO means that the device user does not have to enter any credentials when Web@Work accesses the enterprise server. The Kerberos option is only available if you selected Identity Certificate for Device Authentication.
Server List	Since you typically select <ANY> for the service name for Web@Work, the server list is not applicable. If you do specify service names, enter the internal server's host name or IP address (usually an internal host name or IP address). Include the port number on the internal server that the Sentry can access. For example: sharepoint1.companyname.com:443 You can enter multiple servers. The Sentry uses a round-robin distribution to load balance the servers. That is, it sets up the first tunnel with the first internal server, the next with the next internal server, and so on. Separate each server name with a semicolon. For example: sharepoint1.companyname.com:443;sharepoint2.companyname.com:443.
TLS Enabled	Since you typically select <ANY> for the service name for Web@Work, TLS Enabled is not applicable.

Item	Description
	If you do specify service names, select TLS Enabled if the enterprise servers listed in the Server List field require SSL. NOTE: Although port 443 is typically used for https and requires SSL, the enterprise server can use other port numbers requiring SSL.
Proxy Enabled	Select if you want to direct the AppTunnel service traffic through the proxy server. You must also have configured Server-side Proxy.
Server SPN List	Since you typically select <ANY> for the service name for Web@Work, Server SPN List is not applicable. NOTE: When the Service Name is <ANY> and the Server Auth is Kerberos, the Standalone Sentry assumes that the SPN is the same as the server name received from the device. If you do specify service names, enter the Service Principal Name (SPN) for each server, separated by semicolons. For example: sharepoint1.company.com;sharepoint2.company.com. The Server SPN List applies only when the Service Name is not <ANY> and the Server Auth is Kerberos. If each server in the Server List has the same name as its SPN, you can leave the Server SPN List empty. However, if you include a Server SPN List, the number of SPNs listed must equal the number of servers listed in the Server List. The first server in the Server List corresponds to the first SPN in the Server SPN List, the second server in the Server List corresponds to the second server in the Server SPN List, and so on.

5. Click **Save**.

Configure an AppConnect global policy

Because Web@Work is an AppConnect app, configure an AppConnect global policy. In this policy, you configure AppConnect global settings, which are settings that are not specific to an AppConnect app.

These global settings include:

- whether AppConnect is enabled in the device
- AppConnect passcode requirements
- conditions for wiping AppConnect data
- the default end-user message for when an app is not authorized
- whether AppConnect apps with no AppConnect container policy are authorized by default
- settings for data loss prevention policies
  - For Android devices, these settings apply to all AppConnect apps on the device. Only the Screen Capture setting can be overridden on the AppConnect container policy for Web@Work.

To configure an AppConnect global policy:

1. In the Admin Portal, select **Policies & Configs > Policies**.
2. Select **Add New > AppConnect**.
   If you already have an AppConnect global policy, select it, and click Edit.
3. Fill in the fields as described in "Configuring the AppConnect global policy" in the AppConnect and AppTunnel Guide.
   Most fields default to suitable values, but make sure that you select **Enabled** to enable AppConnect on the device.
4. Click **Save**.
5. Apply the appropriate labels to the AppConnect global policy. If you are using the default AppConnect global policy, it automatically applies to all devices.

Configure an AppConnect container policy for Web@Work

This task is only required:

- If you did not select **Authorize** for **Apps without an AppConnect container policy**, in the AppConnect Global Policy.
- If you want to configure a different set of data loss prevention policies for Docs@Work.

The container policy overrides the corresponding settings in the AppConnect Global Policy.

The AppConnect container policy:

- authorizes an AppConnect app.
- specifies the data loss prevention settings.

NOTE: Ensure that only one Web@Work AppConnect container policy is applied to a device.

To configure an AppConnect container policy for Web@Work:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Select **Add New > AppConnect > Container Policy**.
3. Enter a name for the policy. For example, enter "Web@Work container policy."
4. Enter a description for the policy.
5. In the Application field:
   - For Android, select Web@Work from the dropdown list.
6. Select the settings you require from those described in the following table.

Item	Description
Exempt from AppConnect passcode policy	iOS only
Android Data Loss Prevention	
Allow Screen Capture	Select **Allow Screen Capture** if you want Web@Work to allow screen capture.

7. Select **Save**.
8. Select the Web@Work container policy.

9. Click **More Actions > Apply To Label**.
10. Select the labels to which you want to apply this policy.
11. Click **Apply**.

Enabling Web@Work

The Web@Work configuration is required on the device in order to use Web@Work. It applies to Android devices, and sets up the following features and behaviors:

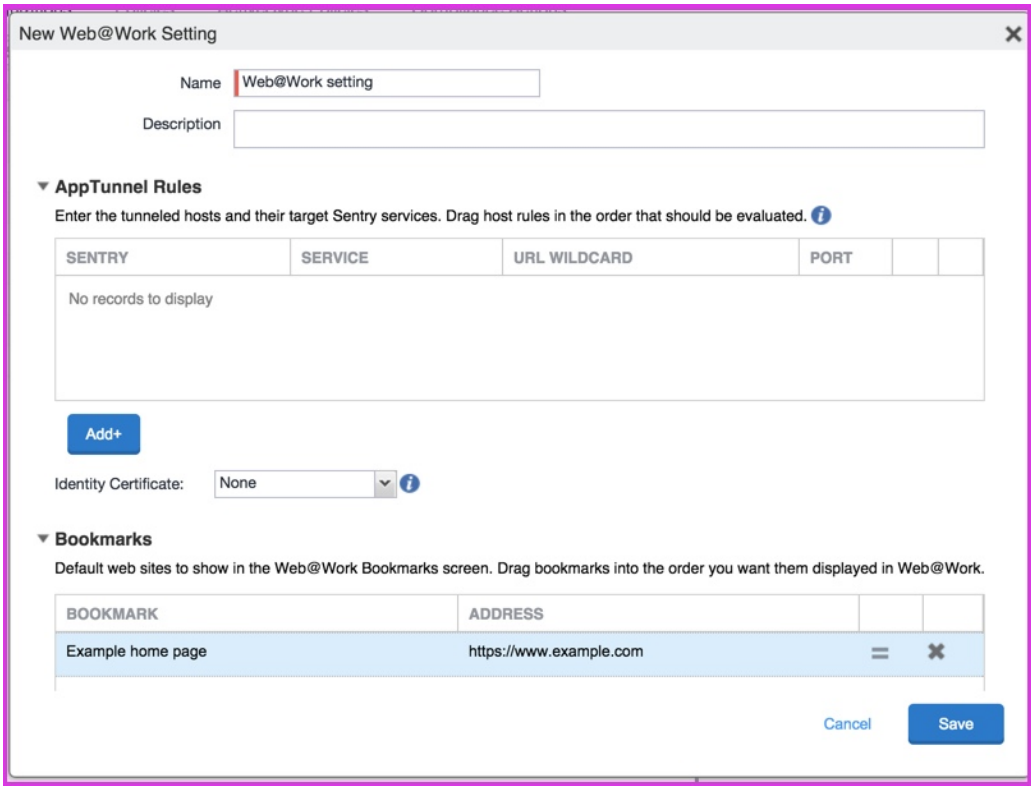
- AppTunnel settings for Web@Work.
  AppTunnel provides secure access to web sites behind your firewall. See Set up a Standalone Sentry to support AppTunnel for Web@Work on page 11.
- Administrator-specified bookmarks.
  The bookmarks you specify here are automatically available to device users.
- Key-value pairs for custom configurations and features.
  Key-value pairs to further customize Web@Work. (These key-value pairs are analogous to the key-value pairs that an AppConnect app configuration provides in its App-specific Configurations section.)

NOTE: Make sure only one Web@Work setting applies to each device.

Configuring a Web@Work configuration

To configure a Web@Work setting:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Click **Add New > Web@Work**.
   The New Web@Work Setting dialog appears.



New Web@Work Setting

Name: Web@Work setting

Description:

AppTunnel Rules

Enter the tunneled hosts and their target Sentry services. Drag host rules in the order that should be evaluated.

SENTRY	SERVICE	URL WILDCARD	PORT		
No records to display					

Identity Certificate: None

Bookmarks

Default web sites to show in the Web@Work Bookmarks screen. Drag bookmarks into the order you want them displayed in Web@Work.

BOOKMARK	ADDRESS		
Example home page	https://www.example.com		

3. Use the following guidelines to create or edit a Web@Work setting:

Item	Description
Name	Enter brief text that identifies this Web@Work setting.
Description	Enter additional text that clarifies the purpose of this Web@Work setting.
AppTunnel Rules Configure AppTunnel rules settings for Web@Work. First, configure the Standalone Sentry to support AppTunnel. See Set up a Standalone Sentry to support AppTunnel for Web@Work on page 11. When Web@Work tries to connect to the URL and port configured here, the Sentry creates a tunnel to the Service.	
Sentry	Select the Standalone Sentry that you want to tunnel the URLs listed in this AppTunnel entry. The drop-down list contains all Standalone Sentrys that are configured to support AppTunnel.
Service	Select a Service Name from the drop-down list. Typically, for Web@Work, the service is <ANY>. NOTE: <CIFS_ANY> is not relevant to Web@Work. This service name specifies an AppTunnel service configured in the App

Item	Description
	Tunneling Configuration section of the specified Sentry. If the service on the Sentry is configured with its Server Auth set to Kerberos, Web@Work uses Single Sign On for the enterprise server. That is, the device user does not enter any further credentials when Web@Work accesses the enterprise app server.
URL Wildcard	Typically, for the Web@Work AppTunnel, enter a hostname with wildcards. The wildcard character is *.. Example: *.yourcompanyname.com If you want finer granularity regarding what requests the Standalone Sentry tunnels, configure multiple AppTunnel rows. If Web@Work requests to access this hostname, the Sentry tunnels the Web@Work data to an app server. The Sentry and Service fields that you specify in this AppTunnel row determine the target app server. Note The Following: • The Web@Work data is tunneled only if Web@Work's request matches this hostname **and** the port number specified in the Port field of this AppTunnel row. If Web@Work requests a hostname that does not match the value of any of the AppTunnel entries in the Web@Work setting, tunneling does not occur. In this case, if the requested hostname is behind your firewall, Web@Work informs the device user that it cannot access the requested hostname. • A hostname with wildcards works only with the service <ANY>. Unlike services with specific service names, these services do not have associated app servers. The Sentry tunnels the data to the app server that has the URL that Web@Work specified. • **The order of these AppTunnel rows matters**. If you specify more than one AppTunnel row, the first row that matches the hostname (and port, for Android) that Web@Work requested is chosen. That row determines the Sentry and Service to use for tunneling. • Do not include a URI scheme, such as http:// or https://, in this field.
Port	Enter the port number that Web@Work requests to access. The Web@Work data is tunneled only if Web@Work's request matches the hostname in the URL Wildcard field **and** this port number. If you do not enter a port number, the port in Web@Work's request is not used to determine whether data is tunneled. NOTE: Entering a port number in this field is required when both of the following are true:

Item	Description
	- The hostname in the URL Wildcard field does not contain a wildcard. - The service is not <ANY>.
Identity Certificate	Select the Certificate or the Certificate Enrollment setting that you created for devices to present to the Standalone Sentry that supports app tunneling. For more information, see “Certificate Enrollment settings” and “Certificates settings” in the MobileIron Core Device Management Guide or Connected Cloud Device Management Guide.
Bookmarks Specify the bookmarks that you want to appear automatically in the Bookmarks screen of Web@Work. The bookmarks appear in the Bookmarks screen of Web@Work in the same order that they appear in the Web@Work setting. To change the ordering, drag the bookmarks in the Web@Work setting.	
Bookmark	Enter the name of the bookmark. The name is any string that describes the URL that the bookmark points to. For example: Sales information
Address	Enter the URL for the bookmark. For example: https://sales.mySecureCompany.com
Custom Configurations Specify Web@Work custom configuration settings as key-value pairs.	
Key	Enter the key. The key is any string that Web@Work recognizes as a configurable item. Unrecognized keys are ignored. See Custom configurations with key-value pairs on page 25 for a description of available keys and values.
Value	Enter the value associated with the key.

4. Click **Save**.
5. Select the new Web@Work setting.
6. Select **More Actions > Apply To Label**.
7. Select the labels to which you want to apply this Web@Work setting.
8. Click **Apply**.

Main configuration steps for Web@Work for Android AppConnect (Cloud)

Following are the main steps for configuring and deploying Web@Work for Android AppConnect on MobileIron Cloud:

1. Go to **Apps > Apps Catalog** and click **+Add.**
2. Select MobileIron Web@Work (Android AppConnect) to add to the Apps catalog.
3. Edit the Category if needed.
4. Enter a brief description of the configuration if needed.
5. Click **Next**.
6. Choose a distribution level for this configuration.
7. Click **Save**.
8. Select **App Configurations** to create app configurations using these options:
   a. Click **Install on Device** to configure the installation or click the **+** icon to add another configuration.
   - Enter a name for the configuration.
   - Optionally add a description for the configuration.
   - Enable or disable the **Install on Device** option.
   - Select the option you want to use.
   - Choose a distribution level for this configuration.
   - Click **Save**.

   b. Click **Promotion** to configure Promotion settings or click the **+** icon to add another promotion configuration.
   - Enter a name for the configuration.
   - Optionally add a description for the configuration.
   - Choose a promotion level for this configuration.
   - Choose a distribution level for this configuration.
   - Click **Save**.

   c. Click **Web@Work Configuration** to configure settings or click the **+** icon to add another Web@Work configuration.
   - Enter a name for the configuration.
   - Optionally add a description for the configuration.
   - Click the **+** icon to add **Bookmarks**.
   - AppConnect Custom Configuration.
   - Click **+Add** to enter Key and Value pairs.
   - Choose a distribution level for this configuration.
   - Click **Save**

   d. Click the **+** icon to set **App Tunnel** options.
   - Enter a name for the configuration.
   - Optionally add a description for this configuration.
   - Enter the domain wildcards for the App Tunnel.
   - Choose a distribution level for this configuration.
   - Click **Save**.

Additional setup steps

The following are optional steps that you can do to further refine the security and access that Web@Work provides.

Compliance actions and security policy

Web@Work is an AppConnect app, and all AppConnect apps are affected by the security policy. You can define the conditions that cause a device to be out of compliance, and configure the security policy to respond to the conditions.

For example, you can create a compliance action that blocks Web@Work from accessing websites that use AppTunnel. If the device becomes non-compliant, the security policy will initiate the compliance action. Compliance actions can also delete (wipe) all Web@Work sensitive data and close its tabs.

For more information, see "Working with security policies" in the MobileIron Core Device Management Guide .

Website authentication using client-side certificates

You can specify client certificates by configuring key-value pairs in a Web@Work setting in the Admin Portal. Two key-value pairs are needed to use this feature:

- one key-value pair for the imported certificate
- one key-value pair for the URL of the website to which you want to present the certificate in response to a challenge

Support of client-side certificates allows users to access internal websites that require certificate-based authentication. The certificate is pushed from MobileIron Core to the device and stored in Web@Work memory.

Limitations

- Web@Work supports one certificate per host.
- Client-side certificates are supported for non-tunneled websites only.

Configuring website authentication using client-side certificates

To configure website authentication using client-side certificates:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.

5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values:

Key	Value Description
`IdCertificate_<number>` Where <*number*> is any positive integer.	The name of the Certificate Enrollment that corresponds to the certificate you want to use.
`IdCertificate_<number>_host` Where <*number*> is the same number you entered for `IdCertificate_`<number>.	The URL for the website to which the certificate will be presented. Wildcards are permitted. Examples: myhost.mycompany.com, *.mycompany.com/myfolder

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Configuring Web@Work for Android AppStation (Cloud)

Deploying a MobileIron unified endpoint management (UEM) platform allows you to secure and manage mobile devices as well as mobile apps in your enterprise. The term mobile device management (MDM) is used to describe the features, policies, and configuration used for securing and managing mobile devices. The term mobile apps management (MAM) is used to describe the features, configuration, and policies used for securing, managing and distributing enterprise apps to mobile devices. In most cases, you deploy MobileIron UEM to do both MDM and MAM. However, in some cases, you may want to manage only the apps on the device without having to manage the device itself. A deployment through which only apps on a device are managed is called MAM-only.

Support for MobileIron Go and MobileIron AppStation on the same device

The support allows for the MAM-only use case where you have contractors whose devices are already managed by another instance of MobileIronCloud, however, you need to deploy required apps from your instance of MobileIron Cloud to the device. The supported MAM-only use cases are described in the MobileIron AppStation for Android Guide for Administrators. The support requires MobileIron AppStation 72 for Android.

MobileIron AppStation is specifically designed as the UEM client for a MAM-only deployment with MobileIron Cloud. The following MAM-only use cases are supported with an AppStation deployment:

- You have employees or seasonal workers who need your relevant apps on their personal devices, but your privacy or legal requirements do not allow device management.
- You have contractors who need your relevant apps, but their devices are managed. The device may be managed by MobileIron Cloud, MobileIron Core or another MDM provider.

MobileIron Cloud allows you to specify Android devices as MAM-only and provides Mobile App Management (MAM) to such devices. A MAM-only deployment allows you to distribute and manage apps without having to manage the device itself. A MAM-only deployment is done through MobileIron AppStation, which is the MobileIron Cloud client for a MAM-only deployment.

NOTE: If you already have a MAM-only deployment using MobileIron Go, you can continue with the deployment. However, MobileIron recommends using MobileIron AppStation for new MAM-only deployments. AppStation supports additional use cases that are not supported with MAM-only using MobileIron Go.

Before you begin

1. Disable all the Android enterprise configurations. For more information, see "Android AppStation and Android enterprise" section in the *MobileIron AppStation for Android Guide*.

   a. In the MobileIron cloud, go to **Configurations > Select the Android enterprise configuration > Edit > Next**.
   b. Deselect the **Enable this configuration** and choose **No Devices**.
   c. Click **Done**.
2. Create a user group to distribute AppStation and manually add users to the group. See "Android AppStation and Android enterprise" in *MobileIron AppStation for Android Guide*. See "Creating a manually managed user group" in the *MobileIron Cloud Administrator Guide*. Create a dynamically managed device group with the rule "user group," which is equal to the user group created for AppStation. See "Adding a device group" in the *MobileIron Cloud Administrator Guide*.

NOTE: Devices previously enrolled for MDM in a Cloud tenant cannot be re-registered with the same Cloud tenant using AppStation. To use AppStation, delete the devices from the Cloud tenant, then register with the same Cloud tenant using AppStation.

3. Verify if MAM Only configuration for AppStation is added and assigned to the dynamically managed device group created for AppStation. See "Configuring MAM-only in MobileIron Cloud" in the *MobileIron AppStation for Android Guide*. Configurations for MAM-only with MobileIron AppStation are created in MobileIron Cloud. MobileIron AppStation receives the configurations when it registers with MobileIron Cloud. Use the MAM Only configuration in MobileIron Cloud to set up AppStation.

Main steps for configuring Web@Work for Android AppStation (Cloud)

The configurations for AppStation are created in MobileIron Cloud. The following provides an overview of the configuration steps for deploying AppStation and pointers to the relevant content in the *MobileIron AppStation for Android Guide*:

- Adding Web@Work for Android AppStation to MobileIron Cloud
- Configuring Web@Work for Android AppStation in MobileIron Cloud

Adding Web@Work for Android AppStation to MobileIron Cloud

Add apps for distribution and assign the apps for distribution to the dynamically managed device group created for AppStation. See "Add apps for distribution." in the *MobileIron AppStation for Android Guide*.

You add Web@Work in the same manner you would add any other Android in-house app. After adding to MobileIron Cloud, you can distribute the app to devices.

Procedure

1. In the MobileIron Cloud, go to **Apps > App Catalog > +Add**.
2. Select **MobileIron Web@Work (Android AppStation)** to add to the Apps catalog.
3. After adding the apps, select the distribution option that includes the users and devices to which you want to make Web@Work for Android available.
4. Click **Next**. If the app was already in the catalog and you are editing the app, click **Save**.
5. Click **Done**.

Configuring Web@Work for Android AppStation in MobileIron Cloud

Following are the main steps for configuring and deploying Web@Work for Android AppStation on MobileIron Cloud:

Procedure

1. In App Configurations for Web@Work select the **Install on Device** and **Promotion** options.
2. Click **Add** to add the Web@Work Configuration.
3. Enter a **Name** for the configuration.
4. Click **+Add Description**, to add text describing the configuration.
5. Click the **+** icon to add **Bookmarks**.
6. In the AppStation Custom Configuration, click **+Add** to enter Key and Value pairs. The key-value pairs for configuring Android AppStation are same as that for Android AppConnect. For more information on key-value pairs, see Configuring custom features using key-value pairs.
7. Choose a distribution level for this configuration.
8. Click **Save**

Registering device to MobileIron Cloud

Notify users to download and register AppStation. If **Always require client registration** is enabled in **Users > User Settings > Device Registration Setting** in MobileIron Cloud, users automatically get emails for registering their device via AppStation. Device users download AppStation to their device directly from the Google Play Store. See "What users see in MobileIron AppStation for Android" in the *MobileIron AppStation for Android Guide* for information about how device users can register their devices to your MobileIron Cloud instance and install apps from AppStation.

Custom configurations with key-value pairs

Key-values pairs are used to provide custom configurations that allow you to manage and control the device user experience.

Configuring custom features using key-value pairs

To add keys and values to your Web@Work setting:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add** to add a new key and value entry.
6. To delete a key-value entry, click the “X” at the right of the key’s row.
7. Click **Save**.

Requesting device camera access (Android)

You can configure Web@Work to request access to the device camera when the device user runs Web@Work. When running Web@Work, a dialog box is shown, requesting user confirmation. After the device user taps the dialog box, Web@Work can access the device camera.

To request access to the device camera:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values:

Key	Value Description

`enable_camera_capture`	`“true” - to enable this feature`
	`“false” - to disable this feature`

7. Click **Save**.
8. Ensure that this Web@Work configuration is applied to labels that include devices that should receive this configuration.

Ignoring warnings from sites with untrusted SSL certificates (Android)

Typically, device users will receive warnings for any websites they attempt to access with untrusted SSL certificates. You can, however, change the default behavior, such that Web@Work will not prompt device users with warnings when browsing sites with untrusted SSL certificates. This is done by configuring one or both of the following key value pairs:

- `trust_all_certificates`: Allows device users to navigate to any website with an untrusted SSL certificate without being prompted by a security warning.
- `trust_certificates_list`: Allows device users to navigate to particular websites with untrusted SSL certificates without being prompted by a security warning.

To enable this feature:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add one or both of the following keys and values:

Key	Value Description

`trust_all_certificates`	`“true” or “yes” - to enable this feature`
	`“false” or “no” - to disable this feature`

`trust_certificates_list`	`“SHA1:NN:NN...NN,SHA256:NN:NN...NN”` For each website, enter a fingerprint value in a comma-separated list.

NOTE: If you set `trust_all_certificates` to false, while entering a comma-separated list of fingerprints for a number of websites, device users will still receive security warnings for any websites outside the comma-separated list.

7. Click **Save**.
8. Ensure that this Web@Work configuration is applied to labels that include devices that should receive this configuration.

Whitelist and blacklist URL filtering (Android)

Using key-value pairs, you can configure a whitelist and/or blacklist of URLs to allow or block access to these URLs from Web@Work.

- A blacklist defines a list of URLs the Web@Work browser is not allowed access to. URLs that do not appear in the blacklist are accessible. If the user navigates to a blacklisted URL, Web@Work displays a warning message. (See also: Custom warning messages for blacklisted URLs (Android) on page 29)

- If both a blacklist and a whitelist are defined, the whitelist rules override the blacklist rules. The whitelist is considered a list of exceptions to the blacklist. The user can always access the URLs that are defined by a whitelist rule.

Blacklist/Whitelist rules and AppTunnel

Blacklist/Whitelist rules do not apply if an AppTunnel is blocked. AppTunnel continues to block the resources as it is configured to do. Users will see the following message: **App tunnel is blocked.**

However, if the AppTunnel is allowed and the resource is blacklisted, then the blacklist is applied. Users will see the following message: **The website you are trying to reach is blacklisted. Please contact your administrator.**

Configuring a blacklist or whitelist

To define a blacklist or whitelist:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values, for a whitelist or a blacklist respectively.

Key	Value Description
`whitelist`	
`blacklist`	

NOTE: URL filter rule strings must be enclosed in double quotes, and comma separated if multiple entries are used.

For “URL_filter_rule”, seeURL Filter Format on page 28.
   - A blacklist can be used with or without a whitelist.
   - A minimum of one URL filter rule must be provided.
7. Click **Save**. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Configuring an exclusive whitelist

An exclusive whitelist allows access to URLs that are in the whitelist, and excludes all other URLs. For example, you may wish to limit Web@Work’s access to only the URLs that belong to your company for a special purpose device.

To configure an exclusive whitelist, use a blacklist value of “*” to match all hosts, and then provide the whitelist URL filter rules to enable access to the desired URLs.

Key	Value Description
`whitelist`	
`blacklist`	`“*”`

URL Filter Format

Web@Work uses the same URL filter format for whitelist and blacklist as does Chromium. See also: www.chromium.org/administrators/url-blacklist-filter-format.

The filter rule can be a URL pointing to a white- or blacklisted resource, or may be a wildcard matching a range of URLs.

The URL filter format is:

```
[scheme://][.]host[:port][/path][@query]
```

Where:

- Scheme is optional, and can be: http, https, ftp, chrome, etc., but cannot be mibrowser or mibrowsers
- Port, path, and query are optional.
- Host is required.
  - Host can be the special value “*”, which matches all hosts.
- If a host is prefixed with a '.' (dot), only exact host matches will be filtered:
  - "example.com" matches "example.com", "www.example.com" and "sub.www.example.com";
  - ".www.example.com" only matches exactly "www.example.com".

Troubleshooting blacklists and whitelists

You can enable debug level of logging for the whitelist and blacklist features by adding the following key-value pair to the Web@Work configuration:

Key	Value Description
`vlog_level`	`“1” -- enables debug level of logs`

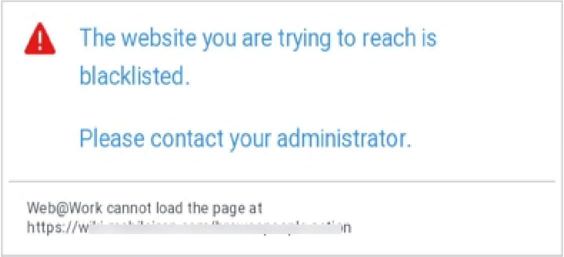
The logs will show the list of allowed and disallowed URLs.

Custom warning messages for blacklisted URLs (Android)

You can provide a customized warning message for blacklisted URLs, using a key-value pair. Device users see the custom warning if they navigate to the affected URLs. This allows administrators to provide additional information about company policies.

When a Web@Work user navigates to a blacklisted URL, they will see the custom warning message instead of the default warning message if the custom warning message is configured

.



The website you are trying to reach is blacklisted.

Please contact your administrator.

Web@Work cannot load the page at https://w[redacted]n

The default text “The website you are trying to reach is blacklisted. Please contact your administrator.” is replaced by your custom text.

TIP: To use the default warning, no configuration is necessary.

Configuring a custom warning message for a blacklist

To add a custom warning message for a blacklist:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following key and value:

Key	Value Description
`blacklist_warning_message`	*`Write your custom message here.`*

Note The Following:

- Do not include double-quotes around the text.
- Text will appear as written. Tag-like content (<, >) and punctuation is shown as-is.

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Secure logout (Android)

You can enable a logout option in the Web@Work menu by using key-value pairs. Set the `enable_logout_button` key-value pair to "true" to enable the logout button.

Logout securely clears all Web@Work browser data. This is especially important if Web@Work is deployed on a shared device that has multiple users. The logout option is a convenient short-cut for users to clear all personal data after a browser session. By logging out of their Web@Work session, the users be certain that all their data has been securely cleared.

Configuring the logout option

To enable this feature:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values:

Key	Value Description

`enable_logout_button`	`“true” - to enable this feature`
	`“false” - to disable this feature`

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Desktop mode (Android)

Using key-value pairs you can allow device users to request the desktop version of a web page instead of the mobile version.

Use the `allow_desktop_mode` key-value pair to enable the desktop mode feature.

Some websites have mobile websites that are less useful than their desktop equivalents. By enabling the desktop mode feature, you allow users to choose to view the desktop version of the site on their mobile devices that would otherwise receive only the mobile version of the site.

Device users will see the option, **Request desktop site,** in the menu for each web page tile.

NOTE: Rendering is up to the website, and not all sites react to this setting.

Configuring desktop mode

To enable desktop mode:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values:

Key	Value Description

`allow_desktop_mode`	`“true” - to enable this feature`
	`“false” - to disable this feature`

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Configuring a home page for Web@Work Android

Using key-value pairs you can specify a URL to be loaded as the home page when Web@Work is launched, or a new tab is opened. Set the `home_page` key-value pair to the desired URL to enable your home page, such as your organization’s internal web portal. Without this setting, the home page is blank with a watermark or lists the configured bookmarks. If this setting as well as bookmarks are configures, the home page presented is a set of links to the bookmarks and the home page configured here.

Configuring a home page for Web@Work

To configure this feature:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following key and supply a URL as the value:

Key	Value Description
`home_page`	

NOTE: There are no quotations around the URL value.

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Enabling form-based auto-fill features (Android)

Administrators can configure whether or not Web@Work saves data and/or passwords that users enter on form-based web pages. By default, saving form data and passwords is enabled.

New menu items in the Web@Work **Privacy** menu enable users to clear saved form data or saved passwords. The administrator can also configure a time limit for saving passwords.

Use the keys `allow_form_autofill, allow_password_autofill,` and `password_autofill_expire` to configure form-based autofill features.

Allowing Web@Work to save data and passwords provides a better user experience when users must enter lengthy authentication credentials, or repetitive form data. You can optionally set a time limit for storing passwords. Credentials can be stored temporarily, for example, during a web session when the user may need to enter their credential several times. The passwords are cleared when the time limit expires.

The user also may clear saved form data and saved password data from the **Privacy** menu in Web@Work. The user also has the option to choose never to save auto-fill data for any given web page.

When form-based auto-fill feature is enabled with the allow_form_autofill key:

- Data on the form is saved after the user submits the data.
- The next time the browser encounters the same form fields, the field is automatically filled with the saved data.
- The user can clear the data by tapping the Web@Work menu, selecting **Privacy > Clear autofill data**, and tapping **Clear Selected**.

If the form-based auto-fill feature is not enabled, form data is not saved. User sees blank form fields even if the form was previously submitted.

When password auto-fill is enabled with the **allow_password_autofill** key:

- When a user enters their credentials on a form-based authentication page, the user is prompted if they want to save their credentials.
- The user can choose **Yes** to save the credentials, or **Never** to never save the credentials for the given site.
- If credentials are saved, the authentication form is auto-populated the next time the user views the authentication page. The users is prompted to save the credentials each time the form is submitted, unless the **Never** option was previously selected.
- If the credentials are not yet saved, or have been cleared, or if the password auto-fill feature is not enabled, the authentication form will be blank.

When the password auto-fill time limit feature is enabled with the password_autofill_expire key:

- The authentication form is auto-populated when a user revisits the form before the time limit expires.
- Password data is automatically deleted after the time limit expires. The next time the user revisits the authentication form, the fields will be blank.
- If the password auto-fill time limit is not set, the saved password remains indefinitely unless it is cleared by the user.

Configuring data and password auto-fill features

To configure the Web@Work form auto-fill features, follow these steps to set the key-value pair described below.

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values:

Key	Value Description

`allow_form_autofill`	`"false"/"no" -- saving form data is disabled`
	`"true"/"yes" -- DEFAULT - saving form data is enabled`

`allow_password_autofill`	`"false"/"no" -- saving password is disabled`
	`"true"/"yes" -- DEFAULT - saving password is enabled`

`password_autofill_expire`	`“nh” -- where n is a number representing # hours to store the password. Example: “8h”`
	`If key is not present (default), password data does not expire.`

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

Clearing saved form data and passwords

Users can clear saved form data and/or saved passwords from Web@Work. On the device:

1. Tap the menu in Web@Work.
2. Tap **Privacy**.
3. Select one or more checkboxes to clear the appropriate data: **Clear Passwords**, or **Clear Autofill Data**.
4. Tap **Clear Selected**.
   The selected data types are cleared from Web@Work.

User agent string for Web@Work (Android)

The user agent for a browser identifies the browser to web server applications, allowing the applications to make choices about the pages and content that they serve. For example:

- For Android, the user agent string for Web@Work on a Nexus 10 tablet running Android 4.2.2 is:
  Mozilla/5.0 (Linux; Android 4.2.2; Nexus 10 Build/JDQ39) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.94 Safari/537.36

Make sure your web server applications handle Web@Work requests just as they would handle native browser requests on the Android device.

Modifying user agent string for Android

The default product name (or user agent) for Web@Work contains “MobileIron/<version-number>”. The presence of the word “Mobile” confuses some web servers, causing them to return only the mobile version of a web page, even on devices with large screens.

Use the `browser_product_name` key to change the product name to “MISecureBrowser” to obviate this behavior.

NOTE: For devices with screens that are 6.5” or smaller, Web@Work automatically uses “Mobile” in the product name.

Use this key-value pair to ensure larger-screen devices will receive appropriate content from web apps or sites that use the term “Mobile” from the user agent to determine content rendering. Small screened devices will continue to render correctly.

Configuring the product name

To configure the Web@Work product name, follow these steps to set the key-value pair described below.

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**.
3. Select the Web@Work setting that applies to the devices of interest.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add**.
6. Add the following keys and values:

Key	Value
`browser_product_name`	“`MISecureBrowser`”

7. Click **Save**.
8. Apply this Web@Work configuration to labels that identify the devices that should receive this configuration.

- Divide for iOS must be installed on the user’s device. <Email+?>

Value	Description
`email+launcher://mibrowser?url=mailto:`	Sets Email+ for iOS as the default app for opening mailto links.
`dividelauncher://mibrowser?url=mailto:`	Sets Divide for iOS as the default app for opening mailto links.

Configuring custom URI formats (Android)

Configuring custom URI formats allows device users to open a custom URI format in web page in Web@Work in a corresponding wrapped app. For example, a URI starting with mailto: can be opened in Email+. You use the key custom_url_scheme to configure custom URI formats.

Web@Work supports the following URI formats by default:

- http://
- https://
- mibrowser://
- mibrowsers://
- file://

NOTE: Web@Work automatically appends http:// if a URI in a valid format does not include a prefix. A URI that is not in a valid format is seen as a search request and is automatically looked up in a search engine.

The above formats do not require any configuration by the administrator. URI formats that are not supported by default result in an error message. Use the custom configuration with key-value pairs to allow users to successfully open custom URIs in a corresponding wrapped app.

To configure Web@Work for Android to open custom URIs:

1. Sign in to the MobileIron Core Admin Portal.
2. Go to **Policies & Configs > Configurations**, select the Web@Work Setting that applies to the devices of interest, and click **Edit**.
3. Under **Custom Configurations**, add a key-value pair.
   - In the Key column, enter `custom_url_scheme`.
   - In the Value column, enter a URI scheme.
     Example: mailto,simple.test,wbx,vnd.youtube
     You can enter a list of URI formats. Each item can be separated by comma (,), semi-colon (;), or pipe (|). Do not include any spaces.
4. Click **Save**.

Configuring Web@Work using Key-Value Pairs (Android)

Certain features in Web@Work for Android can be configured by applying Key-Value Pairs (KVPs) in **Custom Configurations** field in app configuration on Core. The following table lists the features that are configurable using KVPs:

Key	Value	Purpose
Increase logging level		
vlog_level	1	Once set verbose and debug levels of logging are enabled. When enabled, "setting native verbose logging level to 1" line should be shown in logs on Web@Work start.
Access to device camera		
enable_camera_capture	TRUE	Allows Web@Work to ask for access to device camera. Access is granted only after user confirms the proper dialog.
Product name in User Agent string		
browser_product_name	MISecureBrowser	Applied KVP replaces 'MobileIron' word with 'MISecureBrowser' in User Agent string. It allows tablet devices not to be perplexed with 'mobile' word and to display tablet mode.
"Request desktop site" option		
allow_desktop_mode	TRUE	Enables "Request desktop site" option in Web@Work overflow menu.
Configure user certificates		
IdCertificate_1	certificate name	Sets the certificate to be used for authentication. There can be any number in the key - it is used to bind

Key	Value	Purpose
		with IdCertificate_x_host
IdCertificate_1_host	certificate host	Sets the host for authentication with the user certificate
Enable password auto-fill option for sites with both form-based and HTTP-based type of authentication		
allow_password_autofill	• true • yes	Users successfully logged into site with HTTP/form authentication will be suggested to save password for auto-filling (if the site allows it).
	• false • no	Users successfully logged into site with HTTP/form authentication will NOT be suggested to save password for auto-filling (even if the site allows it).
Enable forms auto-fill option		
allow_form_autofill	• true • yes	Values submitted in forms will be suggested in auto-complete list next time users fill this form in.
	• false • no	Values submitted in forms will not be suggested in auto-complete list next time users fill this form in.
Set expiration time for autofill option		
password_autofill_expire		Saved passwords will be available for autofill for the mentioned time in hours. For example, "1", "2h", " 3H ", and so on.
Configure custom home page		
home_page	https://url.url/	URL set for home_page will be loaded in each new tab.

Key	Value	Purpose
Sett white-list and blacklist URL filtering rules		
	"URL_filter_1","URL_filter_2"	URL to which filtering rules are applied will be blocked when accessed.
	"URL_filter_1","URL_filter_2"	URL to which filtering rules are applied will be considered as exceptions from blacklist and will be allowed to access.
Configure custom blacklist warning message		
blacklist_warning_message	Warning message text.	Standard blacklist warning message will be changed to a custom one.
Enable "Logout" button		
enable_logout_button	• true • false	Enables "Logout" button in Settings menu.
Ignore warnings for all sites with untrusted ssl		
trust_all_certificates	• true • yes	Users will NOT be prompted with warnings when navigating to sites with untrusted ssl.
	• false • no	Users will be prompted with warnings when navigating to sites with untrusted ssl, as usual.
Ignore warnings for specific sites with untrusted ssl		
trust_certificates_list	SHA1:XX:XX...XX,SHA256:XX:XX...XX	Users will not be prompted with warnings when navigating to specified sites with untrusted ssl.
Support for URI with custom schemes		
custom_url_scheme	• url_sheme_1 • url_scheme_2	URIs with specified custom

Key	Value	Purpose
		schemes will be treated as URIs in valid format instead of being used as a search request in google search.
Set MixPanel analytics collection ON/OFF		
allow_analytics	• true • false	Administrators can enable or disable analytics collection depending on set value. To disable Mixpanel, enter the following: Key: allow_analytics Value: false **Note: Mixpanel is enabled by default if the key-value pair is not configured.**
Disable Search Engine		

Key	Value	Purpose
	- true - false	When search engine is disabled, Web@Work will treat the terms in the URL navigation bar as web addresses to be resolved. **Note:** By default, search engine is enabled.
Clear cache, browsing history, and other user data		
clear_user_data_after_duration_ in_minutes	15-10080 minutes	You can delete sensitive user data after a defined time period. You can set the value using the *clear_user_ data_after_duration_in_ minutes* key in the admin portal. This feature is disabled by default. Note The Following: - Data is cleared in background when a device locked and time expires. - When the app is ready to delete the data in the foreground mode (timer expires) then the app is sent to background. It waits for 60 seconds before deleting the data in the background mode.

Troubleshooting Web@Work

Clearing browser history and website data

Device users may encounter issues with authentication or page refresh when using Web@Work, as they would in other web browsers. Troubleshooting such issues usually involves clearing the browser history, cookies, and other website data.

You can instruct device users to clear their history and cookies as described here. When managing a device shared by multiple users, it is good practice to instruct users to clear their browser history and cookies for their own privacy.

To clear browser history and website data on an Android device:

1. In Web@Work for Android, tap the menu icon.
2. Tap **Privacy**.
3. Tap the following options:
   - **Clear Cache**
   - **Clear History**
   - **Clear Cookies and Data**
4. Tap **Clear Selected**.
   Web@Work shows a brief confirmation as it clears the data and closes all tabs.

Collecting Web@Work log data

Administrators can enable Web@Work logging by setting key-value pairs in the Web@Work configuration in the MobileIron Core Admin Portal.

This feature allows administrators to view log files generated by Web@Work, making it easier to diagnose and troubleshoot any issues.

Collecting log data for Android devices

For Android devices, use the key vlog_level to enable logging on Web@Work.

To collect and view log data for Web@Work on Android devices:

1. In the Admin Portal, go to **Policies & Configs > Configurations**.
2. Select the Web@Work configuration and click **Edit**.
3. Under **Custom Configurations,** click **Add** to add a key-value pair.

4. Enter the following key-value pair for Web@Work:

Key	Value	Description
`vlog_level`	1	Enables debug level of logs

5. Click **Save**.

The log data will show the list of allowed and disallowed URLs that are configured using the `whitelist` and `blacklist` key-value pairs.