MobileIron Web@Work 2.13.0 for iOS Guide

for MobileIron Core and MobileIron Cloud

March 08, 2021

Contents

New features summary

This guide documents the following new features and enhancements:

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

Web@Work app feature and enhancement

This release includes the following new feature and enhancements.

- **Support for Swedish language enabled**: Web@Work now supports Swedish language. See the **Language support** section under the **Support and compatibility** in the *MobileIron Web@Work for iOS Release Notes*.
- **New indicator "Viewing Mode" added to About screen on iPad devices**: The "**Viewing Mode**" indicates if Web@Work requests Desktop Version or Mobile Version of the websites.

Web@Work administrator features and enhancements

This release does includes the following new administrator features and enhancements.

- **New key-value pair added**: New key-value pair **enable_ipad_desktop_browser** added to enable Web@Work to request desktop version of the websites on iPad devices with iPadOS 13 and later. For more information, see Key-Value Pairs for iOS

Overview of Web@Work for iOS

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 iOS is an AppConnect enabled app.

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. The labeling mechanism in MobileIron Core is used to indicate the devices that 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.

NOTE: On iOS devices, be sure to require a device passcode on the security policy, since a device passcode enables iOS data encryption capabilities. Web@Work uses iOS data encryption capabilities to encrypt browser data.

- 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 a 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 have not already created one to use on 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 on some other AppConnect app.

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

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).
  The device users register Mobile@Work with MobileIron Core by entering their MobileIron credentials. Then, the device user can use Web@Work to access an enterprise app server without having to enter any further credentials. This support depends on your environment being set up to use KCD, plus 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. Also, 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 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.

  Also, 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 iOS

Web@Work for iOS is available to iOS device users in the App Catalog in the MobileIron Core Admin Portal (**Apps > App Catalog**). The app itself is imported from the Apple App Store. The device user uses the Apps@Work web app to discover and install Web@Work from the Apple App Store.

Note that if Web@Work for iOS is installed and launched before the device is registered with Core, Web@Work will run as a unmanaged, standalone app. See AppConnect and non-AppConnect modes for Web@Work for iOS on page 24 for more information.

For information about adding iOS apps to the App Catalog, see “Working with apps for iOS devices” in the MobileIron Apps@Work Guide.

Support and compatibility for Web@Work for iOS

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

About Web@Work for iOS configuration

Web@Work for iOS, like all secure apps for iOS, 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 iOS 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 iOS 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 iOS

When users launch Web@Work for iOS, 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 browsed some web page.
- Options menu: The following options are available when you click the options menu

Option	Function
Copy	Copies the web browser.
Mail	Sends mail to share the web page through email. You should have a configured email account.
Message	Sends the link to the web page through a text message.
Print	Prints the active browser page.
Bookmark	Bookmarks a web page by using this option. This option is enabled only if you have configured the bookmarks option.
Cancel	Minimizes the options menu.

- Bookmarks: The bookmarks option, displays bookmarks if configured by administrators on the core.
- + icon: Opens a new tab.
- Settings: The following options are available:

Options	Functions
About	The About options lists the following information about the Web@Work app: • Version • AppConnect status • Open Source Licenses
Privacy	You can clear user sensitive such as: • Clear History: Displays the browsing history. • Clear Cookies and Data: Clears the cookies and the browsing date.
Passwords & Autofill	• Enable Auto-fill: Saves data such as your 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. • Prompt to save password: Saves passwords for different accounts. • Clear Passwords: Clears the stored passwords.

Web@Work features

Web@Work app for iOS supports 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	iOS	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 Web@Work configuration on page 17
Support for Single Sign On using Kerberos Constrained Delegation (KCD)	iOS	Device users register Mobile@Work with MobileIron Core by entering their MobileIron credentials. They then use Web@Work to access an enterprise app server without having to enter any further credentials. This support depends on your environment being set up to use KCD, plus the necessary AppTunnel configuration. See "Authentication using an identity certificate and Kerberos constrained delegation" in the MobileIron Sentry Guide.
Admin-specified bookmarks	iOS	Web@Work supports bookmarks that you specify on the Admin Portal. **Configuration:** See Web@Work configuration on page 17.
User-specified bookmarks	iOS	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. On iOS, device users can organize their bookmarks so that they display between bookmarks that you specified.
Ability to provide different Web@Work-related settings to different devices and users	iOS	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 Safari	iOS	Because Web@Work uses iOS web technologies, Web@Work automatically inherits any related iOS security updates that are installed on the device.
Ability to delete the cache, browsing history, and the user	iOS	Using a new key-value pair, you can delete sensitive user data after a defined time period. You can set the value using

Web@Work Feature	Platform Support	Description
data such as password and other form-based auto-fill data after a defined time		the *clear_user_data_after_duration_in_minutes* key in the admin portal. The valid range for the key is 15-10080 minutes. If the key value is not in the valid range, the feature is disabled. This feature is disabled by default. Note The Following: • The user data is not deleted when the device is locked and the time expires. • When the app is ready to delete the data in the background mode, it waits for 60 seconds until it deletes the data in this mode.
Ability to perform unified search	iOS	You can search the web using Google. The search feature is disabled by default. To enable the search feature, add the following key-value pair in the Web@Work configuration on MobileIron Core: "enable_search_results_feature_in_addressbar" = YES NOTE: If you are using http tunneling, you must whitelist the Google domain, *google.com*.
Ability to print documents from Web@Work	iOS	You can print documents from Web@Work based on the settings available on AppConnect global policy.
Browser data is encrypted while the device is locked with a passcode	iOS	This data includes the browser cache, HTML5 local storage, cookies, URL history, and bookmarks. NOTE: The security policy must require a device passcode on the iOS device to enable browser data encryption.
Prevent device user from opening a document in another app	iOS	This behavior protects secure documents from leaking to unsecured apps. The behavior is controlled by the AppConnect global policy or AppConnect container policy.
Prevent the device user from pasting into other apps any data that the user copied from Web@Work	iOS	You can choose whether data can be copied from Web@Work to: • no other apps • any other app • only other AppConnect apps This behavior is controlled by the **Copy/Paste To** field of the AppConnect global policy or AppConnect container

Web@Work Feature	**Platform Support**	**Description**
		policy. These restrictions do not impact pasting data *into* Web@Work from other apps. For example, a device user can copy a URL *from* an unsecured app and paste it *into* the Web@Work address bar. Also see: Configure an AppConnect container policy for Web@Work on page 14
URL schemes that open web pages automatically, and only, in Web@Work	iOS	See Apply this Web@Work configuration to labels that identify the devices that should receive this configuration. on page 23
Allow Drag and Drop from Web@Work for iOS 11	iOS	You can drag content from Web@Work for iOS to other AppConnect or third party apps. The drag and drop is controlled by AppConnect Drag and Drop policy. This is disabled by default.
Sharing links from Web@Work	iOS	You can share links from Web@Work app to email clients in the following way: • Using Share button: URL can be shared from Web@Work to an email client using Share button in the tool bar. This behavior is controlled by mailto_prefix and OpenIn policy. • Using Long press: Select the link you want to share, do a long press on the link and the share option is displayed. This feature is supported for email clients for which mailto_prefix key-value pair is configured, such as Email+, Divide, Native, and other AppConnect enabled email clients. • If mailto_prefix key is set: Enables sharing URL to email client based on the value set for mailto_prefix. • If mailto_prefix key is not set: By default native mail client is used to share the link through email. Behavior is based on OpenIn: - If OpenIn = SecureApps: URL sharing is allowed through the native email client. - If OpenIn = Whitelist: If native email client's bundle Id is present in the whitelist, then URL sharing is allowed through the native mail client. Otherwise, sharing not allowed through the native email client. - If OpenIn = All apps: URL sharing is allowed through the native email client. - If OpenIn = Not allowed: Mail option is disabled and

Web@Work Feature	Platform Support	Description
		not appear. This feature is supported for Email+ and native email client.
Importing and Exporting Bookmarks	iOS	You can export user added bookmarks as a file to any other app using the share bookmarks button. The bookmarks file can be imported in Web@Work in any device. The exported bookmark file extension is ***.wwbm***.

Configuring Web@Work for iOS

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

Required configuration for Web@Work for iOS deployment

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

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

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

Main configuration steps for Web@Work for iOS (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.

Import Web@Work for iOS from the Apple App Store

Web@Work for iOS can be imported directly from the Apple AppStore to the App Catalog in the Admin Portal.

To import the app from the Apple AppStore and distribute through Apps@Work:

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.
3. Click **iTunes** to import Web@Work for iOS from the Apple App Store.
4. Enter **MobileIron Web@Work** in the **Application Name** text box.
5. Click **Search**.
6. Select the app from the list that is displayed.
7. Click **Next**.
8. Follow the prompts to add the app.
   The default settings should work in most cases
9. 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 information about adding iOS apps to the App Catalog, see “Working with apps for iOS devices” in the *MobileIron Apps@Work Guide*.

Set up a device passcode (iOS only)

The security policy that you apply to the iOS device requires a device passcode. A device passcode enables iOS data protection, which is necessary for Web@Work to encrypt browser data.

To set up a device passcode on iOS devices:

1. On the Admin Portal, go to **Policies & Configs > Policies**.
2. Select the security policy that applies to the devices that you want to run Web@Work.
3. Click **Edit**.
4. For the **Password** option, select **Mandatory**.
5. Fill in the remaining options relating to passwords.
6. Click **Save**.
7. Repeat steps 2 through 6 for all security policies that apply to devices on which you want to run Web@Work.

For detailed information about security policies, see “Working with security policies” in the MobileIron Core Device Management Guide *for iOS*.

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 for Web@Work for iOS.

Before you begin

Ensure that you have a Standalone Sentry that is set up for AppTunnel and that 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 for iOS, to enable Touch ID authentication to access Web@Work, select **Use Touch ID when Supported** in your AppConnect global policy.
- conditions for wiping AppConnect data
- the app check-in interval for iOS devices, indicating how often AppConnect apps receive AppConnect-related policy updates.
- 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 iOS devices, these settings are used for devices which do not have an 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.

Separate AppConnect container policies are required for iOS and for Android.

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

Procedure

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 iOS, enter `com.mobileiron.securebrowser`.
6. Select the settings you require from those described in the following table.

Item	Description
Exempt from AppConnect passcode policy	iOS only Select this option if you want to allow the device user to use Web@Work without entering the AppConnect passcode or Touch ID. NOTE: When you select this option, situations still occur when the device user must enter the AppConnect passcode or Touch ID. For example, when the user first launches Web@Work, the user is prompted to authenticate.
iOS Data Loss Prevention	
Allow Print	This setting allows an AppConnect app to use print capabilities if the app supports them. However, Web@Work does not allow users to print documents from within Web@Work, even if you select this option.

Item	Description
Allow Copy/Paste To	Select **Allow Copy/Paste To** if you want the device user to be able to copy content from Web@Work to other apps. When you select this option, then select either: • **All apps** Select **All apps** if you want the device user to be able to copy content from Web@Work and paste it into any other app. • **AppConnect apps** Select **AppConnect apps** if you want the device user to be able to copy content from Web@Work and paste it only into other AppConnect apps.
Allow Open In	Select **Allow Open In** if you want Web@Work to be allowed to use the Open In (document interaction) feature. When you select this option, then select either: • **All apps** Select **All apps** if you want Web@Work to be able to send documents to any other app. • **AppConnect apps** Select **AppConnect apps** to allow Web@Work to send documents to only other AppConnect apps. • **Whitelist** Select **Whitelist** if you want Web@Work to be able to send documents only to the apps that you specify. Enter the bundle ID of each app, one per line, or in a semi-colon delimited list. For example: com.myAppCo.myApp1 com.myAppCo.myApp2;com.myAppCo.myApp3 The bundle IDs that you enter are case sensitive. Note The Following: • Web@Work does not allow Open In for these file types: - Text: javascript, ecmascript, css - Application: javascript, x-javascript, json, xml - Image: x-icon - Video: avi, mpeg, mp4, quicktime, H264, x-msvideo - Audio: 3gpp, 3gpp2, iff, x-aiff, amr, mp3, mpeg3, x-mp3, x-mpeg3, mp4, mpeg, x-mpeg, wav, x-wav, x-m4a, x-m4b, x-m4p

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**.

Web@Work configuration

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

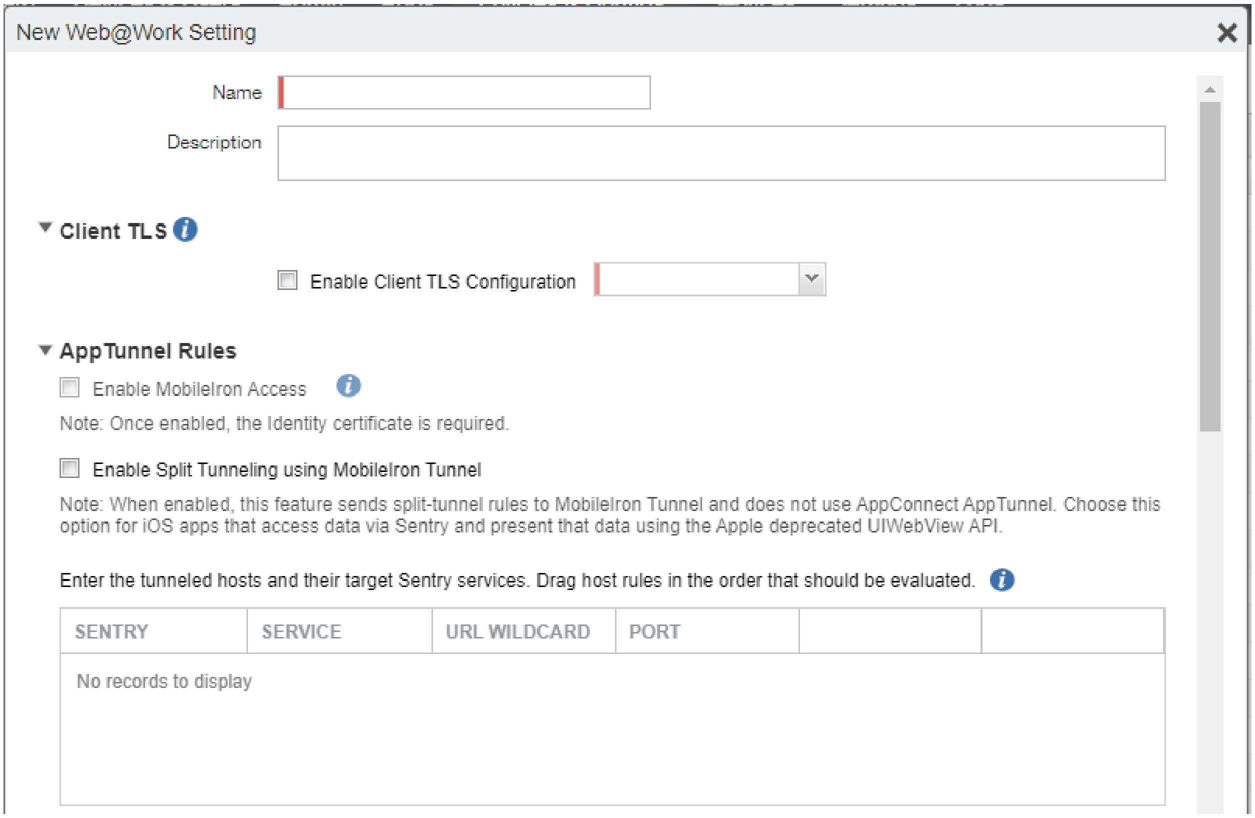
- AppTunnel settings for Web@Work.
  AppTunnel provides secure access to web sites behind your firewall. See Set up a Standalone Sentry to support AppTunnel for Web@Work on page 10.
- Administrator-specified bookmarks.
  The bookmarks you specify here are automatically available to device users.
- key-value 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 Web@Work

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

Description

▼ **Client TLS**

□ Enable Client TLS Configuration

▼ **AppTunnel Rules**

□ Enable MobileIron Access

Note: Once enabled, the Identity certificate is required.

□ Enable Split Tunneling using MobileIron Tunnel

Note: When enabled, this feature sends split-tunnel rules to MobileIron Tunnel and does not use AppConnect AppTunnel. Choose this option for iOS apps that access data via Sentry and present that data using the Apple deprecated UIWebView API.

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					

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.
Client TLS	Enable Client TLS and select the configured Client TLS configuration to use certificate pinning feature to provide more security between Web@Work and enterprise server communication. For more information to configure Client TLS see, *Creating a Client TLS configuration* section in the *MobileIron Core AppConnect and AppTunnel Guide*.
AppTunnel Rules Enable MobileIron Access: Select the checkbox to allow authentication traffic to MobileIron Access. For information about MobileIron Access as a service and how to set up the service with MobileIron Core, see the *MobileIron Access Guide*. 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 10. 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 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

Item	Description
	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 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: • The hostname in the URL Wildcard field does not contain a wildcard. • The service is not <ANY>.
Identity Certificate	Select 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" 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

Item	Description
	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 29 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**.

Locking AppConnect iOS apps when screen is locked

You can lock device users out of AppConnect apps when the device screen is turned due to either inactivity or user action. When locked out of AppConnect apps, the device user must re-enter the AppConnect passcode (or fingerprint) to access AppConnect apps. Locking AppConnect apps when the screen is turned off provides added security to the device. Reasons for the screen turning off include:

- The device user manually turning off the screen, but not locking it. When the screen is off but not locked, the device user can turn on the screen without entering any credentials, such as the device password or fingerprint.
- The device user manually locking the screen.
- The device’s automatic lock timeout expires, as set in the device’s settings.

Select `LockAppConnectContainerOnDeviceLock` option in MobileIron core. Enable the lock, this option is unchecked by default

Main configuring steps for Web@Work for iOS (Cloud)

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

1. Go to Apps > Apps Catalog and click +Add.
2. Select MobileIron Web@Work 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. Select a distribution level.
7. Click Next.
8. Create app configurations using these options:
   a. Click Install Application configuration settings to configure the installation or Click the + icon to add another configuration.
   - Select Install on Device to prompt the user and require installation. This setting uses a silent installation on supervised iOS devices.
   - Select Install as a Managed App. If already installed, it converts the app and its data to a managed app. Converting already installed apps on supervised devices is done silently. The user will be prompted to allow conversion if the device is unsupervised.

   b. Click iOS Application Management configuration settings to configure iOS App Settings or click the + icon to add another configuration.
   - Enter a name for the configuration.
   - Optionally add a description for the configuration.
   - Select Prevent backup to iCloud and iTunes.
   - Select Remove app on unenrollment.

   c. Click Promotion distribution configuration settings 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.

   d. Web@Work configuration
   - Add Bookmarks.
   - AppConnect Custom Configuration.
   - Click +Add to enter Key and Value pairs.

   e. 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.

   f. Click the + icon to add iOS Managed App Configuration settings.
   - Name this configuration.
   - Enter the key and value pairs for iOS 7+Managed App settings.
   - Choose a distribution level.

   g. Choose whether to enable Per App VPN.
   - Name this configuration.
   - Choose a distribution level.

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 *Device Management Guide* specific to your operating system.

Web content filters (iOS only)

Starting with iOS 7, supervised iOS devices support web content filtering. You can set up web content filters to block and allow websites according to your enterprise requirements.

See “Web content filter settings” in the MobileIron Core Device Management Guide or Connected Cloud 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.

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>`	The name of the Certificate Enrollment that corresponds to the certificate you want to use. When the KVP is configured, the certificates are delivered to Web@Work. You do not need to explicitly apply certificate to the label.
`IdCertificate_<number>_host`	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.

Web@Work URL schemes (iOS only)

You can use the following URL schemes to make sure URLs are opened automatically in Web@Work for iOS:

- mibrowser:// for HTTP connections
- mibrowsers:// for HTTPS connections
- mibrowserf:// for full-screen web clips using an HTTP connection
- mibrowsersf:// for full-screen web clips using an HTTPS connection

For example, a web page opens automatically in Web@Work when the device user:

- taps a link in Safari that uses one of these URL schemes.
- taps a web clip that uses one of these URL schemes.

Because iOS otherwise automatically opens HTTP and HTTPS URLs only in Mobile Safari, the native web browser, using these URL schemes in web clips and web pages for mobile devices can improve the user experience when Web@Work is used for tunneling.

Full-screen web clips in Web@Work for iOS

Full-screen web clips allow web apps to be displayed without the browser UI components, such that their look and feel is similar to native iOS apps. Web@Work for iOS enables the same containerization features in full-screen web clips as it does for other web pages, such as copy/paste restrictions, Open In, encrypted browser data, and so on.

For more information about distributing web apps to iOS devices, see the section “Managing Mobile Apps for iOS” in the MobileIron Apps@Work Guide

Situations when Web@Work deletes its sensitive data (iOS only)

Web@Work for iOS deletes (wipes) website data and closes its tabs in the following cases:

- The device is not in compliance and you have specified in the compliance action for the particular non-compliance case to delete data.
- The device user is no longer authenticated with MobileIron Core.

AppConnect and non-AppConnect modes for Web@Work for iOS

Web@Work for iOS can operate in one of two modes:

- AppConnect mode, where the app is managed by MobileIron Core
- non-AppConnect mode (or “standalone” mode), where the app is not managed by MobileIron Core.

For Web@Work to be in AppConnect mode, the following conditions must be met:

- The device must have Mobile@Work installed and must be registered to MobileIron Core when Web@Work is launched for the first time.
- On Core, the “Authorize Apps without an AppConnect container policy” option must be selected (checked) in the AppConnect global policy, or there must be a valid AppConnect container policy.

When Mobile@Work is already installed on the device and registered to MobileIron Core, installing or updating Web@Work from the App Store puts Web@Work into AppConnect mode when Web@Work is launched for the first time.

Web@Work will be in a non-AppConnect, unmanaged mode if:

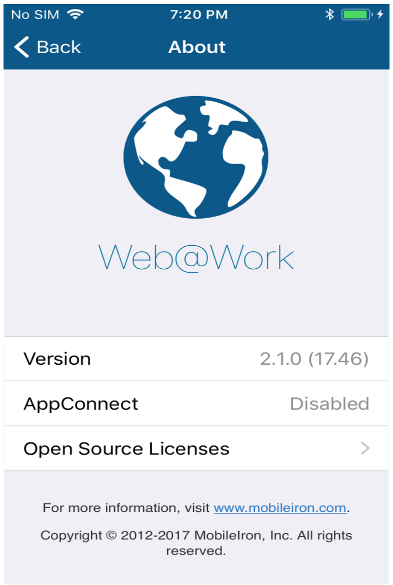
- Web@Work is installed and launched before the device is registered with MobileIron Core.

Web@Work installed in this manner always remains in non-AppConnect mode (unmanaged by MobileIron), even if you later install Mobile@Work on the device.

Verifying that Web@Work for iOS is running as an AppConnect app

To determine whether Web@Work for iOS is operating as an AppConnect-enabled app or a non-AppConnect app:

1. In Web@Work on the iOS device, tap the **Settings** icon.
2. Tap **About**.
3. Check the value of the **AppConnect** setting, which indicates if AppConnect is "enabled "or "disabled".



Version 2.1.0 (17.46)

AppConnect Disabled

Open Source Licenses

For more information, visit www.mobileiron.com.

Copyright © 2012-2017 MobileIron, Inc. All rights reserved.

To change Web@Work mode to AppConnect mode

The only way to change the Web@Work mode from AppConnect disabled to AppConnect enabled is to remove and reinstall Web@Work:

1. Remove Web@Work from the device.
2. Install Mobile@Work for iOS and register the device with Core.
3. Reinstall and launch Web@Work.

NOTE: If Web@Work is launched for the first time when there is no AppConnect container policy and "Apps without an AppConnect container policy" is not selected in the Authorize checkbox of the AppConnect Global Policy, Web@Work enters standalone mode and is not blocked with the "app is unauthorized" error message.

Split Tunneling using MobileIron Tunnel

Due to Apple deprecation of support for **UIWebView** and the impact that has on AppConnect AppTunnel on iOS, there is a new option, **Enable Split Tunneling using MobileIron Tunnel** in the AppTunnel configuration for Web@Work on MobileIron unified endpoint management (UEM) platform.The MobileIron UEM platforms are MobileIron Cloud or MobileIron Core.

Before enabling the option in MobileIron UEM, ensure that MobileIron Tunnel is deployed and the Tunnel VPN configuration is applied to the Web@Work for which you are enabling the split tunneling option.

Enabling the split tunneling option allows the tunnel rules to be applied to MobileIron tunnel for Web@Work. The new feature is introduced due to the planned deprecation of the UIWebView API by Apple.

NOTE: The Web@Work configuration for split tunneling overrides the MobileIron Access configuration for split tunneling, this does not impact the other apps that use MobileIron Access configuration.

In addition to MobileIron Tunnel 4.1.0, the feature requires either one of the following:

- Mobile@Work 12.3.0 and MobileIron Core 10.7.0.0.
- MobileIron Go 5.4.0 and MobileIron Cloud 70.

For information about configuring AppConnect App Configuration and AppTunnel configuration on MobileIron Cloud, see "Configuring AppConnect Apps" and "Configuring AppTunnel traffic rules" sections in the MobileIron Cloud Administrator Guide.

For information about configuring AppConnect App Configuration on MobileIron Core, see "AppConnect app configuration" in the *MobileIron Core AppConnect* and *AppTunnel Guide*.

The feature requires **Mobile@Work 12.3.0** and **MobileIron Tunnel 4.1.0 for iOS**. For information about the UIWebView API deprecation, see UIWebView Deprecation and AppConnect Compatibility.

Configuring split tunneling with MobileIron Tunnel (Core)

This section describes the steps to configure split tunnel on Web@Work.

Before you begin

- Ensure that MobileIron Sentry service is active. For more information, see Enabling split tunneling section in the *MobileIronAccess Guide*.
- Ensure that MobileIron Tunnel is deployed and a Tunnel VPN configuration is applied to the AppConnect app. For information about deploying MobileIron Tunnel for iOS, see the *MobileIron Tunnel for iOS Guide for Administrators*.

Adding Per App VPN to Web@Work app

The following steps describe how to add Per App VPN to Web@Work configuration. Ensure that Per App VPN profile is already created.

Procedure

1. In the Admin Portal, go to **Apps > App Catalog**.
2. Click Web@Work, click **Edit**.
3. Under the **Per App VPN Settings**, select **Per App VPN by Label Only** checkbox.
4. Select the VPN available in the list and click the right arrow.
5. Click **Save**.

Editing Web@Work Configuration

The following steps describe how to edit Web@Work configuration to enable Split Tunneling on MobileIron Core.

Procedure

1. In the **Admin Portal**, go to **Policies & Configs > Configurations**.
2. Select the check box for Web@Work configuration.
3. Click **Edit**, in the **Edit Web@Work Setting** page, go to AppTunnel Rules.
4. Under the **AppTunnel Rules** section, select the **Enable Split Tunneling using MobileIron Tunnel** option.
5. Click **Save**.

For information about configuring AppConnect App Configuration, see "AppConnect app configuration" in the *MobileIron Core AppConnect and AppTunnel Guide*.

For more information Creating Per App VPN or MobileIron Tunnel VPN setting, see VPN settings in the *MobileIron Core Device Management Guide for iOS and macOS Devices*.

Configuring split tunneling with MobileIron Tunnel (Cloud)

This section describes the steps to configure split tunnel on Web@Work for MobileIron Cloud.

Before you begin

- Add and configure MobileIron Tunnel app. For more information, see Main tasks for configuring MobileIron Tunnel for iOS (Cloud) section in the *MobileIron Tunnel for iOS Guide for Administrators* guide.
- Ensure that you have a Standalone Sentry set up for AppTunnel and the necessary device authentication is also configured. See “Configuring Standalone Sentry for app tunneling” in the *MobileIron Sentry Guide*.
- Ensure Per App VPN is created.

Editing Web@Work configuration

The following steps describe how to edit Web@Work configuration to enable Split Tunneling on MobileIron Cloud.

Procedure

1. In the MobileIron Web@Work **App Configuration > AppTunnel**, click **+ icon**.
2. Enter the **Name** of the configuration.
3. In the **App Tunnel** section, edit the following fields:
   a. Sentry Profile

   b. Turn **ON** the **Enable Split Tunneling using MobileIron Tunnel** option.

4. Add App Tunnel rules.
5. Choose a distribution option for the configuration.
6. Click **Save**.
7. In **App Configuration > Per App VPN** and click **+ icon**.
8. Enter the **Name** of VPN configuration.
9. Select the **Enable Per-App VPN for this app** check-box to select MI Tunnel configuration from the drop-down list.
10. Choose a distribution option for the configuration and click **Done**.
11. Click **Save**.

After configuring split tunneling, ensure that the configurations are pushed to the device.

For more information, see https://help.mobileiron.com/s/article-detail-page?Id=kA12T000000TTetSAG.

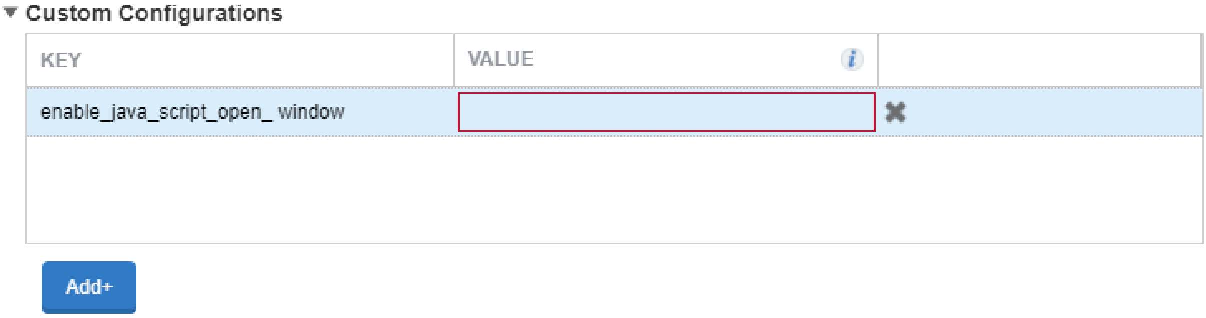
Custom configurations with key-value pairs

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 your device.
4. Click **Edit**.
5. Under **Custom Configurations**, click **Add** to add a new key and value entry.



KEY	VALUE
enable_java_script_open_ window	

6. To delete a key-value entry, click the “X” at the right of the key’s row.
7. Click **Save**.

Configuring a home page for Web@Work

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

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.

NOTE: For iOS, you can also use the key allow_passwords_autofill.

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. The 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.

Enabling secure form autofill on pages where auto-complete is disabled (iOS only)

To enable form autofill on pages where auto-complete is disabled:

1. In the MobileIron Core 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 key `respect_form_autocomplete_attribute`.
5. Assign the key a value of `yes` or `true`.
6. Click **Save**.

User agent string for Web@Work

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 iOS, the user agent string for Web@Work on an iPad running iOS 7.0.4 is:
  Mozilla/5.0 (iPad; CPU OS 7_0_4 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Mobile/11B554a MobileIron/1.3.0 Version/7.0.4 Safari/537.51.1

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

WKWebView limitations

WkWebView is the default rendering engine. The WKWebView support has the following limitations:

- AppTunnel feature provided by AppConnect does not work with WKWebView. The alternative is to use split tunneling. For more information on split tunneling, see Split Tunneling using MobileIron Tunnel.
- Certificate based authentication works for devices running on iOS12 and higher.
- Cookies storage remains in unencrypted format.
- Cookies might get deleted when you upgrade from previous version of Web@Work which is using UIWebView.

- Complete cache storage is not encrypted.

Configuring mailto links in Email+(iOS)

You can configure Web@Work for iOS to open “mail to” links in Email+ for iOS in a Web@Work setting in the MobileIron Core Admin Portal.

The organization maintains a higher level of security by configuring mailto links tapped in Web@Work to open in a secure email app.

Before you begin

- Web@Work must be AppConnect enabled.
- Divide for iOS must be installed on the user’s device.

Configuring Web@Work to open mailto links in Email+ or Divide

To configure Web@Work for iOS to open mailto links in Email+ for iOS:

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.
4. In the Key column, enter `mailto_prefix`.
5. In the Value column, enter one of the following values:

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

6. Click **Save**.
7. Select the labels to which you want to apply this Web@Work setting. Apply the labels to the relevant devices.

NOTE: If mailto key-value pair is not added in the Web@Work configuration, then the mailto links will open in Native mail application and not as per the Open-In DLP policy.

Document sharing between Docs@Work and Web@Work

Document sharing is available between Docs@Work (verion 2.7.0 and iOS 11.0 and higher) and Web@Work. This means that you can access documents from Docs@Work and use Web@Work to upload the documents to any web page.

To configure this feature you need to enable the following key-value pairs.

- **MI_ENABLE_DOC_SHARING**: Enables document sharing in Docs@Work.
- **MI_AC_ACCESS_CONTROL_ID**: Controls access to the documents present in Docs@Work. If this key-value pair is not present in Docs@Work configuration, then Docs@Work documents are accessible to all the apps.
  Add this key-value pair to both Docs@Work and Web@Work configuration, to allow access to only Web@Work.
- **MI_AC_DOCUMENT_EXTENSION_DLP**: Supports documents sharing between Web@Work and Docs@Work in an unencrypted format.
  To enable this, set **MI_AC_DOCUMENT_EXTENSION_DLP = "ALL"** in the Docs@Work configuration.
- **MI_SHARED_GROUP_ID**: Controls file sharing between a subset of AppConnect apps. The value for this KVP has to be same when used across AppConnect apps. This is an optional KVP.

Configuring document sharing from Docs@Work to Web@Work

To share documents from Docs@Work to Web@Work for uploading:

1. Set `MI_ENABLE_DOC_SHARING = YES` in Docs@Work configuration.
2. Set `MI_AC_ACCESS_CONTROL_ID = <string>` KVP on both Web@Work and Docs@Work configuration to access the documents. The value should be set to the same string for both Docs@Work and Web@Work configurations.
3. Set `MI_AC_DOCUMENT_EXTENSION_DLP = "ALL"` in Docs@Work configuration.

NOTE: Web@Work 2.5.0 supports document sharing only from Docs@Work to Web@Work in an unencrypted format.

NOTE: When the MI_AC_DOCUMENT_EXTENSION_DLP = "All" key-value pair is set in Docs@Work configuration, then the documents are shared in plain text to all apps passing the access control including Email+ and any other AppConnect apps.

NOTE: MI_AC_DOCUMENT_EXTENSION_DLP key-value pair is not mandatory when Docs@Work and Email+ are configured for document sharing and if the key-value value is missing, documents shared from Docs@Work are AppConnect encrypted and Web@Work does not support AppConnect encrypted documents for uploading.

Key-Value Pairs for iOS

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

Key	Value	Description
Enable resubmitting credentials during NTLM authentication		

Key	Value	Description
NtlmAuthRetryOnIos8OrNewer	1	Set to avoid repeated prompts during NTLM authentication for iOS 8 devices and newer devices.
Configuration of 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 with IdCertificate_x_host.
IdCertificate_1_host	Certificate host	Sets the host for authentication with the user certificate.
Credentials auto-fill		
allow_passwords_autofill	• true • false **or** • yes • no	Allows to disable credentials auto-fill, which is enabled by default if this KVP is not set.
allow_password_autofill	• true • false **or** • yes • no	Allows to disable credentials auto-fill, which is enabled by default if this KVP is not set (just renaming of KVP). This KVP should be used for Web@Work 2.0 and higher.
allow_form_autofill	YES	Form-based autofill is disabled by default, use allow_form_autofill : YES
respect_form_autocomplete_attribute	• true • false **or** • yes • no	Allows autofilling credentials from login forms where autocomplete=“off” is set. In KVP it should be set to false or no.
Send feedback functionality		
feedback_email_address	your_email@xxxx.com	Enables sending feedback and sets email address where feedback should be sent
log_files_limit	Any number	Sets the number of log files that are created. Put "0" for unlimited. Default value is 10.

Key	Value	Description
use_emailplus_application_for_feedback	- true - false	Enables sending feedback via Email+. Default value is false.
AppConnect logs		
MI_AC_LOG_LEVEL	- Error - Info - Verbose - Debug	Specifies the level of logging from the least to the most verbose.
MI_AC_LOG_LEVEL_CODE	Any string	Underspecification prompted in Mobile@Work to activate AppConnect logs.
MI_AC_ENABLE_LOGGING_TO_FILE	- Yes - No	Enables collecting AppConnect logs to a file in Web@Work.
Customized user-agent and web-kit version		
webkit_version	- "537.51.2" to simulate iOS 7 - "600.1.4" to simulate iOS 8	Specifies web kit version for web-pages.
user_agent	user agent string	Specifies user agent string for web-pages. If both KVPs are set, user_agent overrides webkit_version. For Example: Mozilla 5.0 (iPhone; CPU iPhone OS 7_1 like Mac OS X) AppleWebKit/537.51.2 (KHTML, like Gecko) Mobile/11D167 MobileIron/10.0 Version/7.1 Safari/600.1.4
Set Divide/Email+ as default app to open mailto links		
mailto_prefix	email+launcher://mibrowser?url=mailto: dividelauncher://mibrowser?url=mailto:	Sets Email+ as default app to open mailto links. Sets Divide as default app to open

Key	Value	Description
		mailto links. You can share a URL through an email client.
Disable strict SOP		
strict_same_origin_policy	• true • false **or** • yes • no	Disables enforcing strict Same Origin Policy. Default value is "true"
Enabling/disabling copy and paste for Service Provider		
forbid_loading_about_blank	• yes • no	Enabling copy/paste for specific service provider. Disable skipping of loading of about:blank for child elements Enables possibility to paste copied text into specific form on Service Provider -> Profile. Default value is "no". Disables skipping of loading of about:blank for child elements.
Set homepage		
home_page	URL, for example: http://www.yahoo.com	Sets homepage for each new tab.
Set expiration period for autofill		
password_autofill_expire	Xh, for example: 4h	Sets expiration period for auto filled credentials. h - is optional Applicable for both login forms and http based authentication.
Enable custom keyboards		
MI_AC_IOS_ALLOW_ CUSTOM_KEYBOARDS	• true • false	true: allows the use of custom keyboards

Key	Value	Description
		false: does not allow the use of custom keyboards. Default if key-value is not configured: true.
Skip percent encoding of ";" in URL Path		
skip_percent_encode_for_semicolon_in_URL	• true • false **or** • yes • no (case insensitive)	Skip percent encoding of ";" character in URL Path component. Default value is yes
Handle DOM Mutations after Initial document load		
handle_DOM_mutations_after_intial_document_load	• true • false **or** • yes • no (case insensitive)	Handles Document Object Model (DOM) mutations like adding "new child" or perform attribute change after initial document load. Default value is yes.
Defer JavaScript location changes for child window		
defer_javascript_location_changes	• true • false **or** • yes • no (case insensitive)	Defer JavaScript location changes for child window until first argument of window. Open call is completely loaded in child window. Default value is yes.
Disable Window Body Unload event listener		
disable_window_body_onunload_event_listener	• true • false **or** • yes • no (case insensitive)	Disables window body unload event listener for the window and for all the frame windows in the page. Default value is yes.
Remove sensitive user browser data (History, cache, Form-data, cookies, Pasteboard, saved-password)		

Key	Value	Description
clear_user_data_after_duration_in_minutes	Value in minutes	Remove user's sensitive data after certain time interval as specified by the user from MobileIron Core. By default the functionality is disabled. Valid range is 15 - 10080 (in minutes). Any out-of-range value disables this feature.
Ignore errors while loading any internal resources and embedded frames in a web page		
ignore_errors_in_resources_and_embedded_frames	• true • false **or** • yes • no (case insensitive)	Setting "ignore_errors_in_resources_and_embedded_frames" KVP to YES will ignore any errors while loading internal resources or embedded frames and will not result in a complete page error. The default value is no.
Inject FastClick javascript library in Web@Work for all web pages		
inject_fastclick_js_library	• true • false **or** • YES • NO (case insensitive)	By default FastClick javascript library injection is disabled, and the value is set to "NO" If the value for this key is set to "YES", it will inject FastClick javascript library for all the web pages being loaded in Web@Work.
Enable search results feature in Web@Work address bar		
enable_search_results_feature_in_addressbar	• true • false **or** • YES • NO (case insensitive)	Setting "enable_search_results_feature_in_addressbar" to "YES" enables the search results feature in the address bar. If this feature is enabled and you

Key	Value	Description
		type some word in the address bar (which is not a URL) and then press Enter, Web@Work will show search results from search engines such as Google. If this feature is disabled Web@Work will consider the typed string as URL and will try to load the page.
Enable resetting of scaling of web page after document loading completes		
enable_resetting_scale_to_fit_for_scaling_webpage	• true • false **or** • YES • NO (case insensitive)	The default value is "YES". If the value for this key is set to "NO" there will be no resetting of scaling of a web page after document loading completes.
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.
browser_product_name	MISecureBrowser	Allows you to configure product name in user agent string. If the key-value pair is not present in the Web@Work configuration , then the user agent string will have "MobileIron" as the product name by default. The value for the key "browser_product_name will replace the

Key	Value	Description
		string "MobileIron" from the user agent string.
Miscellaneous		
MI_AC_USE_ORIGINAL_ COOKIES_FOR_DOMAINS	The value of the KVP must be comma-separated domain names. Do not put in any spaces.	Allow Web@Work to send custom cookies in web requests: Some web pages inject custom cookies into web requests. For example, when an end user taps on a link in a web page, the page's JavaScript injects a custom cookie. If a user makes such a request from a web page displayed in Web@Work, by default AppConnect does not include the injected cookies in the web request, which can cause the request to fail. AppConnect now includes the custom cookies in the request if the MobileIron server administrator includes the following key in the Web@Work's app-specific configuration on the MobileIron server: MI_AC_USE_ ORIGINAL_COOKIES_FOR_ DOMAINS. The value of the key is a comma-separated string listing the domains for which the custom cookies should be included. Make sure no spaces are included in the value. For example: www.somewebsite.com, somename.someotherwebsite.co m

Key	Value	Description
enable_java_script_open_window	- true - false	Allows Web@Work to enable JavaScript pop-up window.
enable_ipad_desktop_browser	- true - false	Allows Web@Work to request desktop version for all the websites on iPadOS 13 devices and later.

Troubleshooting Web@Work

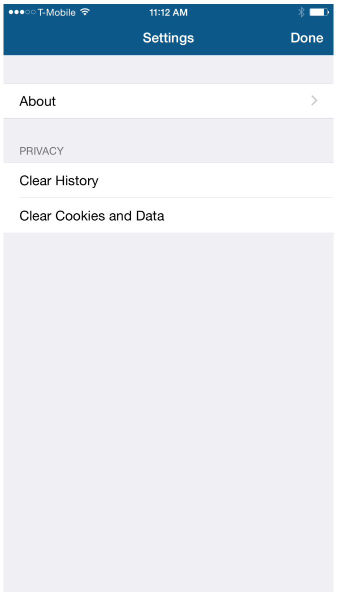
Device users may encounter issues with authentication or page refresh when using Web@Work, as they would in other web browsers. Troubleshooting such issues usually involves clearing the browser history, cookies, and other website data.

Clearing browser history and 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 iOS device:

1. In Web@Work for iOS, tap the **Settings** icon.



2. Tap either or both of the following:
   - **Clear History**
   - **Clear Cookies and Data**

   Web@Work shows a prompt requesting confirmation of the action you tapped.
3. Tap **Clear** to delete browser history or cookies and other website data.
4. Tap **Done**.

   Web@Work shows a confirmation prompt before clearing this data. After tapping to confirm, all tabs are closed.

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 iOS devices

For information about using AppConnect log data, see the section "Logging for AppConnect apps for iOS" in the AppConnect and AppTunnel Guide.

To collect and view log data for Web@Work on iOS 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 pairs for Web@Work:

Key	Value	Description
`log_files_limit`	0-10	This key limits the number of log files generated by Web@Work. The default value is 10. A value of 0 means there is no limit to the number of log files that can be generated. Each log file is 256KB.
`feedback_email_address`	*the administrator's email address*	This key configures the email address to which the Web@Work log files are sent. It also creates a Feedback button in Web@Work on users' devices. Device users tap the Feedback button to send logs to the email address associated with the `feedback_email_address key.`

5. Click **Save**.

Using Web@Work app on a non-compliant device

In some cases the Web@Work app might retire, in that case the Web@Work app should be uninstalled and re-installed from MobileIron Apps@Work. The Web@Work app retires when,

- the compliance policy action is set to Quarantine, the Web@Work app retires on a non-compliant device. Even when the device returns to a compliant state, the Web@Work app remains in retired state.

- the "Wipe AppConnect data after x days" is configured in Global AppConnect policy, the Web@Work app retires if the device is not connected to the server within the ‘ x’ days. Even when the device establishes connection, the Web@Work app remains in retired state.
- the Web@Work app or AppConnect is disabled from the server and later enabled, the Web@Work app remains in retired state.