ivanti

Guide

CONTENTS

1. Ivanti Email+ 5.0.0 - 5.5.0 for iOS Guide **3**
2. Revision history **4**
3. New features summary **5**
4. Overview of Ivanti Email+ for iOS **10**
4.1. About Ivanti Email+ for iOS 11
4.2. About Ivanti Email+ for iOS configuration 12
4.3. What users can do in Email+ for iOS 13
5. Configuring Ivanti Email+ for iOS **14**
5.1. Required components for an Email+ for iOS deployment 15
5.2. Before you configure Ivanti Email+ for iOS 16
5.3. Main steps for Configuring Ivanti Email+ for iOS (Ivanti EPMM) 17
5.4. Main steps for configuring Ivanti Email+ for iOS (Ivanti Neurons for MDM) 24
5.5. Ivanti Email+ installation on an iOS device (Ivanti EPMM and Ivanti Neurons for MDM) 26
5.6. Ivanti Email+ configuration field description (Ivanti Neurons for MDM) 27
6. Additional configurations using key-value pairs **30**
6.1. Key-value pairs for customization Ivanti Email+ for iOS 31
6.2. Key-value pairs for customization Ivanti Email+ for iOS (Cont.) 43
6.3. S/MIME support in Ivanti Email+ for iOS 54
6.4. Background email checks and user notifications 57
6.5. Configuring Ivanti Web@Work for iOS to open mailto links in Ivanti Email+ for iOS 58
6.6. Allow copy from Ivanti Email+ for iOS to other AppConnect apps only 59
6.7. Email and calendar classification capabilities (Deprecated) 60
6.8. Classification markers 62
6.9. Crash reporting capabilities 67
6.10. Allow logging on Ivanti Email+ 68
6.11. Suppress certificate email check 69
7. Real-time push notifications **70**
7.1. About real-time push notifications for Email+ for iOS 71
7.2. Standalone Sentry setup for real-time push notifications 75
7.3. Deployment use cases for real-time push notifications 78
7.4. Before you configure real-time push notifications 79
7.5. Overview of configuration on Ivanti EPMM 81
7.6. Description of configurations in Ivanti EPMM 83
7.7. Overview of configuration on Ivanti Neurons for MDM 85
7.8. Description of configurations in Ivanti Neurons for MDM 87
7.9. Keys for real-time and interval-based push notifications (Ivanti EPMM and Ivanti Neurons for MDM) 90
7.10. Verifying that the Ivanti Neurons for MDM notification service is working 95
8. Troubleshooting Ivanti Email+ for iOS **96**
8.1. Setting up logging for Email+ for iOS (Ivanti EPMM) 97
8.2. Detailed logging for AppConnect apps for iOS (Ivanti EPMM) 99
9. What users see **100**

1. Ivanti Email+ 5.0.0 - 5.5.0 for iOS Guide

September 2026

Ivanti Email+ for iOS is an AppConnect-enabled app. AppConnect containerizes apps to protect content on iOS and Android devices. Each AppConnect app becomes a secure container whose content is encrypted and, protected from unauthorized access.

New features summary

Revision history

2. Revision history

Date	Revision
September 03, 2026	Updated for Ivanti Email+ 5.5.0 release.
January 21, 2026	Updated for Ivanti Email+ 5.2.0 release.
November 18, 2025	Updated for Ivanti Email+ 5.1.0 release.
August 26, 2025	Updated for Ivanti Email+ 5.0.0 release.

3. New features summary

These are cumulative release notes. If a release does not appear in this section, then there were no associated new features and enhancements.

Ivanti Email+ 5.5.0 New features summary

Ivanti Email+ app features and enhancements

- **Support for split view and window mode on iPad**: Email+ on iPad now supports Split View and Window Mode, aligning with the iOS 26 deprecation of the UIRequiresFullScreen option. The app seamlessly transitions between regular and compact display modes using UISplitViewController across all screens, with the calendar leveraging a dedicated root view controller for each mode. This enhancement ensures full multi-tasking support on iPad with a consistent UI/UX experience during mode switches.
  UISplitViewController will be used for all screens except the calendar.

Ivanti Email+ administrator features and enhancements

- **New value added**: New value **SHA-256**, **SHA-384**, **SHA-512** is added to **email_signing_digest** key-value pair as approved by SafeLogic. New value **AES-256**, **AES-192**, **AES-128** is added to **email_encryption_algorithm** key-value pair as approved by SafeLogic.
- **Migration from Exchange Web Services to Microsoft Graph API: Email+ is transitioning Microsoft 365 connectivity from Exchange Web Services (EWS) to the Microsoft Graph API. Users can continue to use core email and calendar features with Microsoft 365 accounts. However, delegated and shared account functionality may behave differently under Microsoft Graph due to Microsoft Graph API capabilities, permissions, and service-specific requirements.**
- **KVP added for Microsoft Graph support**: Email+ now supports Microsoft Graph API for delegated mailbox, shared mailbox, delegated calendar, and shared mailbox calendar functionality in Microsoft 365 environments. A new feature flag **graph_api** is added to enable Microsoft Graph API.
  The existing KVPs for Microsoft Graph API:
  - email_graph_host
  - msal_graph_resource_scope
  - msg_min_allowed_auth_mode
  - realtime_notifications_protocol

> Before enabling the graph_api feature flag and configuring the Microsoft Graph KVPs, ensure that Email+ version 5.5.0 or later is installed on user devices. Enabling this feature on earlier versions is not supported.

For more information, see **Key-value pairs for customization Ivanti Email+ for iOS (Cont.)**and **Migration from Exchange Web Services to Microsoft Graph**.

Ivanti Email+ 5.4.0 New features summary

Ivanti Email+ app features and enhancements

- **New section added for Calendar invite**: The following updates are made for Calendar events:
  - The mail list has a new section with a meeting request. It shows the event details directly from the mail list.
  - The RSVP button opens the event details.
  - The Calendar snippet (one hour before/after window) added to the event details to show available events before and after the received invite (for invites only).

Support enabled for the Liquid Glass interface: The Email+ app has now adopted Apple's new Liquid Glass interface offering a dynamic, responsive UI where colors, elements, and visuals adapt to light, motion, and your screen's context for a more immersive experience.

Deprecated iOS 16: Support for iOS 16 is now deprecated from Email+ 5.4.0.

Ivanti Email+ administrator features and enhancements

This release does not include new administrator features and enhancements.

Ivanti Email+ 5.2.1 New features summary

Ivanti Email+ app features and enhancements

- **Integrated iOS Appconnect 5.5.1**: The Email+ app is now integrated with iOS AppConnect 5.5.1.

Ivanti Email+ administrator features and enhancements

This release does not include new administrator features and enhancements.

Ivanti Email+ 5.2.1 New features summary

Ivanti Email+ app features and enhancements

- **Integrated iOS Appconnect 5.5.1**: The Email+ app is now integrated with iOS AppConnect 5.5.1.

Ivanti Email+ administrator features and enhancements

This release does not include new administrator features and enhancements.

Ivanti Email+ 5.2.0 New features summary

Ivanti Email+ app features and enhancements

- **Enhanced notification process for emails stuck in Outbox**: You can now view a red warning icon if an email is stuck in the Outbox for more than 7 days. You will see a dialog box once per day which states "*Some emails have been stuck in the Outbox for over 7 days. Open the Outbox to send or delete them.*"
- **Integrated Microsoft Teams with Email+ Calendar**: The Email+ app calendar now supports Microsoft Teams integration. For more information, see *Microsoft Teams integration with Email+Calendar* in the Email+iOS Guide.

Ivanti Email+ administrator features and enhancements

- **New KVP added to enable Microsoft Teams meeting**: The admin can now enable Teams Meeting feature with the **microsoft_teams_meetings** key value pair. For more information, see *Key-value pairs for customization Ivanti Email+ for iOS Cont* in the Email+iOS Guide.
  - **New KVPs added for Graph API calls**: The following KVPs must be added to enable Graph API in Email+5.2.0 release.
    - **email_graph_host**: The KVP **email_graph_host** has been added with the value graph.microsoft.com' to enable the Microsoft Teams feature. When configuring Microsoft Graph, administrators must use graph.microsoft.com as the default Exchange host. Users may need to use different endpoints depending on their national cloud deployments. For further details and a comprehensive list of supported Microsoft Graph endpoints, please refer to the official documentation: **Supported Microsoft Cloud Endpoints**.
    - **msal_graph_resource_scope**
    - **msg_min_allowed_auth_mode**

  For more information, see *Key-value pairs for customization Ivanti Email+ for iOS Cont* in the Email+iOS Guide.
  To enable Microsoft Graph API support and Teams integration, administrators must grant admin consent to the Email+ Auth Application within their Azure tenant. For more information, see **Microsoft Graph API Support**.

Ivanti Email+ 5.1.0 New features summary

Ivanti Email+ app features and enhancements

- **Support to import multiple attachments from AppConnect or third party applications**: You can now share or import multiple attachments between AppConnect or third-party apps to Email.

> This feature is not supported for Email+ Share.

- **Support to save embedded images added:** You can now directly save the embedded or inline images from an email in the Email+ app to files or other apps.
- **Ability to send auto-reply to external users**: You can now set an out of office message for external users by selecting "**Auto-reply to external users**" option. You can have different message for internal and external recipients.
- **Ability to add attachments to event supported**: When creating an event from the Email+ app, you can now add attachments to the event.
- **Improved Contacts search**: Users can now search contacts using any word or letters present in the contact name, even if the first or last name consists of multiple words.
- **Support to set Email+ as the default email application**: You can now set Email+ as the default email app on your iOS device. In your iOS device, go to **Settings > Apps > Email+ > Default Email App**.
- **Ability to add a display name for Delegated or Shared accounts**: In the Email+ app, under the **Calendar tab > All Calendars screen**, the delegated or shared accounts now have a **Display Name**.
- **Support to copy contact fields**: You can now copy contact fields from Contact details page.
- **Support to preview attachments**: You can now preview attachments while composing an email.
- **Improved delegated and shared account functionality**: You can now add a Shared or Delegated mailbox or calendar from **Settings > Accounts** section.

Ivanti Email+ administrator features and enhancements

- **Improved the allow_logging key-value pair**: If an MSAL related error occurs during authentication, you can now share the Email+ logs using the "**Collect Logs**" button displayed in the error alert. For more information, see Key-value pairs for customization Ivanti Email+ for iOS (Cont.).
- **New value added**: New **ai_tools** value added to the **enabled_features** key-value pair to enable Apple Intelligence controls. For more information, see Key-value pairs for customization Ivanti Email+ for iOS (Cont.)

Ivanti Email+ 5.0.0 New features summary

Ivanti Email+ app features and enhancements

- **Email+ is now migrated to Microsoft Authentication Library**: The Email+ app is now migrated from Microsoft Azure Active Directory Authentication Library (ADAL) to Microsoft Authentication Library (MSAL) to support Microsoft Conditional Access. The Email+ users using Sentry in Active Sync Mode should use Sentry version 9.20.0 and later. For more information, see ***Ivanti Email+Customers using Sentry must upgrade their Sentry version 9.20 or above***.
- **Tab bar position change starting iPadOS 18**: The position of the tab bar is moved from the bottom of the screen to float over content at the top of the screen.
- **Azure Conditional Access**: The Email+app now supports Azure Conditional Access and Passwordless Authentication. For more information, see **Configuring Azure Conditional Access (Optional)**

Ivanti Email+ administrator features and enhancements

- **New KVP for Web View in Email+ Configurations**: The admin can now use optional KVP **msal_internal_web_view** to switch from Safari to internal web view for Email+ configurations with **Per-App VPN**. It is useful if Safari disrupts VPN during authentication. For more information, see **Key-value pairs for customization Ivanti Email+ for iOS (Cont.)**
- **email_address key-value pair is used for authentication**: The **email_address** key-value pair value is now used for authentication instead of **email_exchange_username**. This change is part of the Microsoft Authentication Library (MSAL) which is now used instead of the deprecated Microsoft Azure Active Directory Authentication Library (ADAL). Thus, email_address and email_exchange_username should have same values for upgrade to Email+ 5.0.0.
- **New KVP added for Conditional Access:**: The following KVPs are added to Email+ 5.0.0 release.
  - **msal_client_id**: Allows replacing the default application ID used by MSAL (MicrosoftAuthentication Library) with a custom value.
  - **msal_resource_scope**: Defines the resource scopes required by MSAL (Microsoft AuthenticationLibrary) to acquire access tokens for protected resources. Multiple scopes should be provided as a single comma-separated string. For example, user.read, mail.read, and so on.
  - **msal_redirect_URL**: Allows URL redirection for MSAL.

    > These KVPs are optional and required only if the customer has Email+ Azure application in their Azure tenant.

- **New KVP added to disable Microsoft Authenticator launch for authorization**: The admin can now disable Microsoft Authenticator launch with **msal_disable_authenticator_use** KVP.
- **New KVP added for troubleshooting**: Two new KVPs are added to configure additional logging for troubleshooting.
  - **allow_logging_insecure**: Allows detailed logging with sensitive details. Enable only if requested by Ivanti for issue debugging. For more information, see **Key-value pairs for customization Ivanti Email+ for iOS (Cont.)**
  - **allow_logging_responses**: It allows to record requests and responses. Enable only if requested. For more information, see **Key-value pairs for customization Ivanti Email+ for iOS (Cont.)**

4. Overview of Ivanti Email+ for iOS

Ivanti Email+ for iOS provides secure email, calendar, contacts, notes, and tasks on corporate-owned and BYOD iOS devices by communicating with an ActiveSync server in your enterprise.

The following provide an overview of the Email+ app for iOS devices:

4.1. About Ivanti Email+ for iOS

Email+ for iOS is an AppConnect-enabled app. AppConnect containerizes apps to protect content on iOS and Android devices. Each AppConnect app becomes a secure container whose content 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 content. AppConnect apps are managed using policies configured in a Unified endpoint management (UEM) platform. The UEM platform is either **Ivanti EPMM** or Ivanti Neurons for MDM.

As an AppConnect app, all Email+ content is secured. The app interacts with other apps according to the data loss prevention policies that you specify. The app has the following secure features:

- **Secure apps passcode**: A secure apps passcode, if you require one, protects access to all secure apps. This is the AppConnect passcode, which you define in UEM. The AppConnect passcode provides an additional layer of security for secure apps, beyond the device passcode.
- **Data encryption**: AppConnect encrypts all AppConnect-related data on the device, such as Email+ app data, app configurations, and policies. This means app data is secure even if a device is compromised.
- **Data loss prevention**: You determine whether Email+ for iOS can use the iOS copy/paste, open-in, and open-from features. AppConnect data loss prevention policies control if users can copy/paste data out of Email+ and control how email attachments can be shared with other apps via open-in and open-from.

For information about AppConnect features and configuration beyond Email+ for iOS, see the AppConnect and AppTunnel Guide.

Where to find Ivanti Email+ for iOS

Email+ can be downloaded using enterprise Apple App Store. Email+ for iOS, like all secure apps can only be distributed as an in-house app. When you distribute Email+, distributing the Secure Apps Manager is required.

You can make Email+ available to device users as an in-house app in the App Catalog in the Ivanti EPMM Admin and Ivanti Neurons for MDM (under Apps > App Catalog > In-House). The device user launches Mobile@Work to discover and install Email+, where it will appear under Secure Apps within the Mobile@Work app.

4.2. About Ivanti Email+ for iOS configuration

You configure settings for Email+ in the UEM platform. Because UEM provides these settings to the app, device users do not have to manually enter configuration details. By automating the configuration for device users, each user has a better experience when installing and setting up the app. Also, the enterprise has fewer support calls, and the app is secured from misuse due to configuration.

These settings include, for example:

- the ActiveSync server, or the Standalone Sentry that interacts with the ActiveSync server.
- the user ID for the ActiveSync server.
- the SCEP or certificate setting for the certificate that the device presents to the Standalone Sentry for authentication, if you are using certificates for authentication.
- Kerberos Constrained Delegation with Standalone Sentry, which provides a better user experience for device users.

4.3. What users can do in Email+ for iOS

When users launch Email+ for iOS, users can do the following from the main screen:

- **Email**: Send and receive their corporate email, and manage any sub-folders.
- **Calendar**: Manage and synchronize their corporate calendar data, including meetings and appointments in a daily, monthly, or list view.
- **Contacts**: Manage and synchronize their corporate contacts.
- **Notes**: Manage, synchronize, and create new notes.
- **Tasks**: Manage, synchronize, and create new tasks.
- **Settings**: Manage their certificates, keys, recognized certificate authorities, as well as alerts, sync period, S/MIME signing and encryption, and so on.

5. Configuring Ivanti Email+ for iOS

You can configure Email+ for iOS using different components. It can be configured on both Ivanti EPMM and Ivanti Neurons for MDM.

The following describe how to set up Email+ for iOS:

5.1. Required components for an Email+ for iOS deployment

The following components are required for an Email+ for iOS deployment:

- Unified endpoint management (UEM) platform: Ivanti EPMM or Ivanti Neurons for MDM
- Sentry, with ActiveSync enabled (required if you want to secure access to the ActiveSync server using Sentry)
- An iOS device that is registered with a UEM
- Client application: Mobile@Work for Ivanti EPMM deployments, Go for Ivanti Neurons for MDM deployments.

For supported versions see, Email+ for iOS Release Notes.

A device user who launches Email+ for iOS without UEM platform will be running Email+ for iOS as an unsecured standalone app during a 30 day trial.

> If a device user has already launched Email+ for iOS as a standalone trial app, the device user must uninstall and reinstall Email+ for iOS to use it as a secure AppConnect-enabled app.

5.2. Before you configure Ivanti Email+ for iOS

Before you configure Email+ for iOS:

- Ensure that all devices to which you plan to deploy Email+ must be able to access https://activate-emailplus.mobileiron.com. This URL enables the use of ActiveSync features in Email+. No identifiable information, however, is reported to the server.
- If you are using Sentry to allow access to your enterprise ActiveSync server, you must have either an Integrated Sentry (Ivanti EPMM only) or Standalone Sentry installed and configured for ActiveSync with the necessary device authentication set up.
- For related documentation see the following:
  - For information on how to install Standalone Sentry, see the Ivanti Sentry Installation Guide
  - For information on how to set up Standalone Sentry for ActiveSync, see Ivanti Sentry Guide for Ivanti Neurons for MDM OR"Adding an entry for Standalone Sentry in Ivanti EPMM" in the Ivanti Sentry Guide for Ivanti EPMM
  - For information on how to set up Integrated Sentry, see "Adding an entry for Integrated Sentry in Ivanti EPMM" in the Ivanti Sentry Guide for Ivanti EPMM.
- For Ivanti Neurons for MDM, you will configure the Exchange (ActiveSync) service as part of the Standalone Sentry setup for ActiveSync.
- In an Email+ deployment separate Exchange configuration is not required. The Email+ configuration contains the necessary settings. For more information on Configuration see, **Configuring Email+ for iOS (Ivanti EPMM)** and **Configuring Email+ for iOS (Ivanti Neurons for MDM)**. For more information on key-value pairs, see **Additional configurations using key-value pairs**.

5.3. Main steps for Configuring Ivanti Email+ for iOS (Ivanti EPMM)

This section explains the steps to configure Email+ for iOS Ivanti EPMM.

> You do not configure a separate Exchange setting for the device as you do for other email apps. The AppConnect app configuration provides the necessary information.

Adding Ivanti Email+ for iOS to Ivanti EPMM as a recommended app

Device users can download Email+ for iOS directly from the Apple App Store. You can also distribute Email+ for iOS as a recommended app through Apps@Work.

Procedure

1. In the Admin Portal, go to Apps > App Catalog.
2. From the Quick Import drop-down list, select iOS.
3. Enter Email+ in the Application Name text box.
4. Click Search.
5. Select the app from the list that is displayed.
6. For Email+, click Import.
7. Click OK on the pop-up message, and close the Quick Import dialog.
8. Email+ is now listed in the App Catalog. Information included in the app, such as the name, is automatically configured. All other settings, such as the App Category and whether the app is a free app, are set to default settings.

> To view and edit the settings for the app, click on the app name in the App Catalog.

1. Select the app to apply the app to a label:
    1. Click Actions > Apply to Label.
    2. Select the label that represents the iOS devices for which you want the selected app to be displayed.
    3. Click Apply.

Next steps

Continue to **Enabling third-party AppConnect apps**.

Related topics

- For more information on adding iOS apps to the app distribution library, see “Working with apps for iOS devices” in the Apps@Work Guide. See also, “Setting per app VPN priority” in the Ivanti EPMM Apps@Work Guide.
- For information on creating a VPN setting, see the Ivanti Tunnel for iOS Guide.

Enabling third-party AppConnect apps

Email+ for iOS requires you to enable the licensing option for third-party and in-house AppConnect apps.

Procedure

1. In the Admin Portal, go to Settings > System Settings.
2. Click Additional Products > Licensed Products.
3. Select AppConnect For Third-party And In-house Apps.
4. Click Save.

Next steps

Continue to **Configuring the AppConnect global policy**.

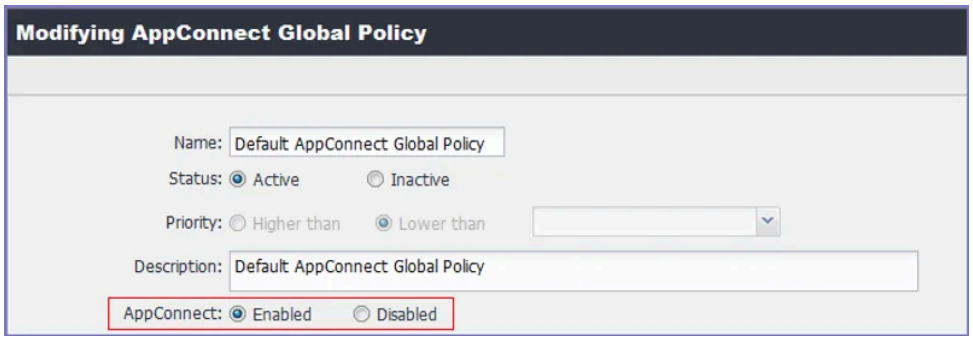
Configuring the AppConnect global policy

Because Email+ for iOS is an AppConnect app, AppConnect must be enabled in the AppConnect global policy if it has not yet been configured. The AppConnect global policy specifies AppConnect app settings such as AppConnect passcode and data loss prevention requirements. You can use the Default AppConnect Global Policy.

> Most fields are set to suitable default values.

Procedure

1. In the Admin Portal, go to Policies & Configs > Policies.
2. Select the Default AppConnect Global Policy and click Edit.
3. Enable AppConnect in the AppConnect global policy



1. For AppConnect, select Enabled.
2. (Optional) Scroll down to the Data Loss Prevention Policies section.
3. (Optional) For Apps without an AppConnect container policy, select Authorize.

> If you do not select this option, then you must create an AppConnect container policy for Email+.

1. (Optional) If you select Authorize for Apps without an AppConnect container policy, also select the data loss preventions options you want to enable.
2. Click Save.

> If you create a new AppConnect Global Policy, you must apply it to the appropriate labels. You do not need to apply the Default AppConnect Global Policy to a label.

Procedure: Applying to a label

1. Select the AppConnect global policy.
2. Click More Actions > Apply To Label.
3. Select the appropriate labels to which you want to apply the policy.
4. Click Apply.

Next steps

Continue on **Configuring the AppConnect container policy**.

Related topics

For more information about the AppConnect Global policy, see the "Configuring the AppConnect global policy" section in the Ivanti AppConnect for Ivanti EPMM Guide and Ivanti Tunnel for iOS Guide for detailed description of each field.

Configuring the AppConnect container policy

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 Email+.

The AppConnect container policy authorizes an AppConnect app and specifies the data loss prevention settings. The container policy overrides the corresponding settings in the AppConnect Global Policy.

> Make sure to apply only one AppConnect container policy for Email+ for iOS.

Procedure

1. In the Admin Portal, select Policy & Configs > Configurations.
2. Select Add New > AppConnect > Container Policy.
3. Enter a name for the policy.
4. Enter a description for the policy.
5. In the Application field, enter the bundle ID for the app:
   com.mobileiron.ios.emailplus
6. Configure the iOS data loss prevention policies according to your requirements.
7. Click Save.
8. Select the container policy.
9. Select More Actions > Apply To Label.
10. Select the labels to which you want to apply the policy.
11. Click Apply.

Next steps

Continue on to **Creating an AppConnect app configuration for Ivanti Email+**.

Creating an AppConnect app configuration for Ivanti Email+

Email+ for iOS requires an AppConnect app configuration in Ivanti EPMM. The AppConnect app configuration provides the type of information that is usually configured in an Exchange setting, such as the fully qualified domain name and user ID for the ActiveSync server, and certificate information. As such, Email+ for iOS does not require an Exchange setting.

The AppConnect app configuration for Email+ for iOS also includes the bundle ID for the app and key-value pairs used to configure app settings.

Make sure to apply only one AppConnect app configuration for Email+ for iOS to each device.

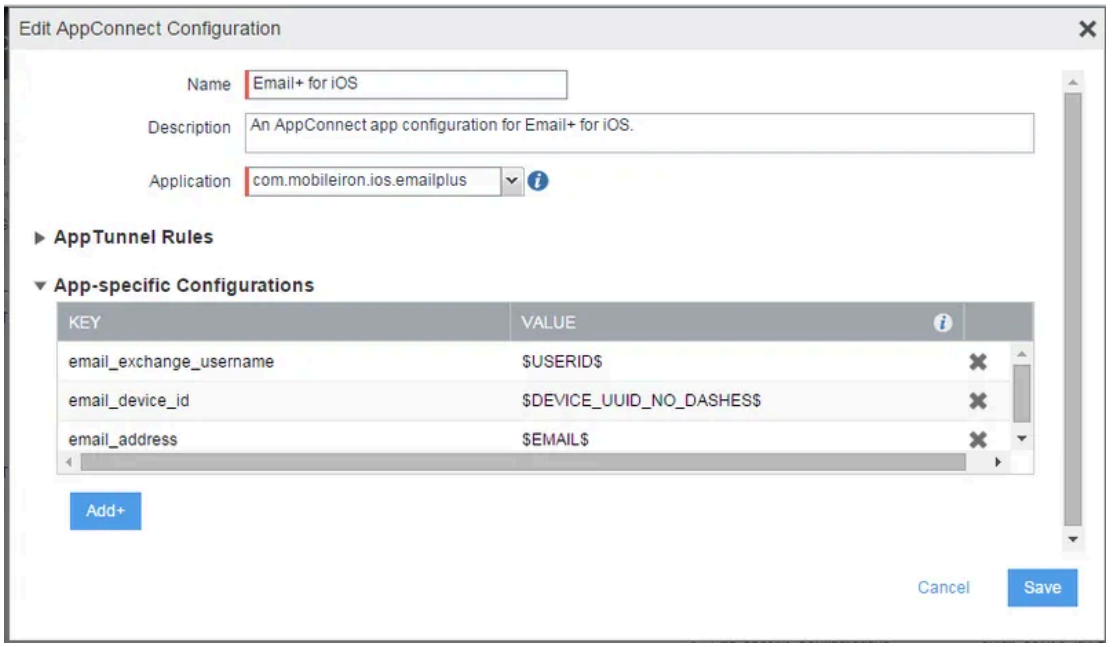
> If you make a mistake in the configuration, the app shows a message to the device user indicating an error in configuration.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policy & Configs > Configurations.
2. Click Add New > AppConnect > Configuration to create a new AppConnect configuration.
3. In the Name field, enter brief text that identifies this AppConnect app configuration.
4. Example: Email+ for iOS
5. In the Description field, enter additional text that clarifies the purpose of this AppConnect app configuration.
6. In the Application field, enter the bundle ID for the app: com.mobileiron.ios.emailplus
7. In the App-specific Configurations section enter the following required key-value pairs:

Key	Value
email_exchange_host	Fully qualified domain name of your ActiveSync server or Sentry.
email_ssl_required	Enter true to secure communication using https to the server that you specified in email_exchange_host. Otherwise, enter false. Typically, set this field to true unless you are working in a test environment.

8. Click Save.
9. When you save an app configuration with the bundle ID com.mobileiron.ios.emailplus, Ivanti EPMM automatically applies the following key-value pairs to the app configuration:
10. email_exchange_username with value $USERID$
11. email_device_id with value $DEVICE_UUID_NO_DASHES$
12. email_address with value $EMAIL$



10. Select the new AppConnect app configuration.
11. Click More Actions > Apply To Label.
12. Select the labels to which you want to apply the AppConnect app configuration.
13. Click Apply.

ActiveSync server synchronization due to app configuration changes

Email+ for iOS synchronizes all emails, contacts, calendar, and task items with the ActiveSync server when the device user first launches Email+ for iOS. It also does a full synchronization if you change the values of the following keys in the app configuration:

- email_address
- email_exchange_host
- email_exchange_username

After you have changed one of these values, the full synchronization occurs the next time Email+ for iOS receives the updated app configuration. Email+ for iOS receives the update the next time it runs after the AppConnect app checkin interval has expired.

> The first Email+ for iOS synchronization with the ActiveSync server may require considerable time and bandwidth, as does changing the values of the keys mentioned here.

Customize Ivanti Email+ app behavior with key-value pairs

Administrators can customize Email+ app behavior by configuring key-value pairs in the App-specific Configurations section of AppConnect app configuration for Email+ for iOS. These key-value pairs define app behavior such as providing detailed notifications to device users and export contacts from Email+. See **Additional configurations using key-value pairs** for the complete list of custom key-value pairs.

Configuring email attachment control with Standalone Sentry

With Email+ for iOS, you can configure Standalone Sentry to deliver emails with attachments to the secure app. The attachments can then only be shared with other apps according to your data loss prevention policies.

Therefore, when using secure email apps, you typically configure Standalone Sentry to use the email attachment control setting called Open With Secure Email App.

Procedure

1. Go to Settings > Sentry in the Ivanti EPMM Admin Portal.
2. Select the Standalone Sentry that handles email for the devices.
3. Click the edit icon.
4. In the section Attachment Control Configuration, select Enable Attachment Control.
5. For iOS And Android Using Secure Email Apps, select Open With Secure Email App.
6. Click Save.

Related topics

- For more information about email attachment control, see the Ivanti Sentry Guide for Ivanti EPMM.

Informing users to install Ivanti Email+ for iOS

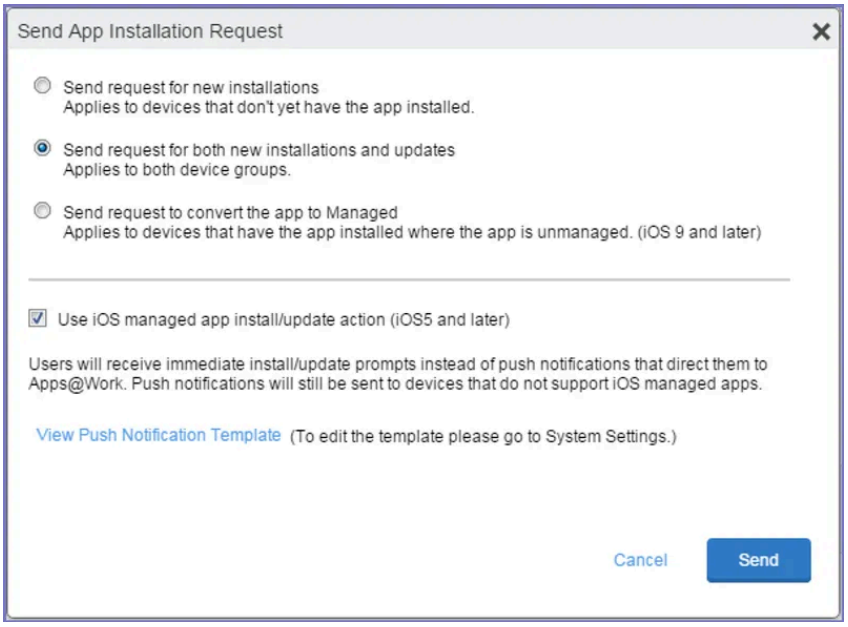
You can inform device users by sending an APNS (Apple Push Notification Service) notification that directs device users to the new or updated Email+ for iOS app in Apps@Work. Or, you can send an installation request directly to all devices in the labels applied to Email+ for iOS, bypassing Apps@Work entirely.

As with badge notifications, updates are determined by comparing the version number of the installed app with that of the update.

> The notification feature applies only to apps designated as Featured apps.

Procedure

1. In the Admin Portal, go to Apps > App Distribution Library.
2. Select iOS from the Select Platform list.
3. Select the featured app you want to work with.
4. Click Message.Send app installation request.



5. Use the following guidelines to select the app installation option:

Item	Description
Send request for new installations	Prompts the device user to install the app if it is not already installed.
Send request for updates	Prompts the device user to update the app if it is not already updated.
Send request for both new installations and updates	Prompts the device user to install or update the app.
Use iOS managed app install/update action	Ignore the Apps@Work display and immediately install or update the app.

6. To check the content of the message prior to sending:
7. a. Select the Push Notification template from the list.
8. b. Click View Messages.
9. Click Send.
10. The message is sent only for apps configured as featured apps in the app distribution library.

5.4. Main steps for configuring Ivanti Email+ for iOS (Ivanti Neurons for MDM)

You add Ivanti Email+ from the Ivanti Neurons for MDM app catalog, and as part of the setup, you also specify the app configurations.

Following are the main steps for configuring Email+ for iOS on Ivanti Neurons for MDM:

Adding Ivanti Email+ for iOS on Ivanti Neurons for MDM

Email+ for iOS is available in the app catalog in Ivanti Neurons for MDM.

Procedure

1. In Ivanti Neurons for MDM, go to Apps > App Catalog > +Add.
2. In Business Apps, click Email+ (iOS).
3. Make any updates as necessary and click Next
4. You can change the category and add a description.
5. Choose a distribution option for the app and click Next.
6. Update the default install settings or add install settings as necessary.
7. Update the promotion settings or add promotion settings as necessary.
8. For Email+ configuration, click + to add an Email+ configuration.

Next steps

- Configuring Ivanti Email+ for iOS on Ivanti Neurons for MDM.

Configuring Ivanti Email+ for iOS on Ivanti Neurons for MDM

The Email+ configuration provides the type of information that is usually configured in an Exchange setting, such as the fully qualified domain name and user ID for the ActiveSync server, and certificate information. As such, Email+ for iOS does not require an Exchange setting. The configuration for Email+ for iOS also includes the bundle ID for the app and key-value pairs used to configure app settings.

> Make sure that only one Email+ for iOS configuration is distributed to a device.

> If you make a mistake in the configuration, the app shows a message to the device user indicating an error in configuration.

Procedure

1. In the Email+ configuration, enter a name for the configuration.
2. Configure the Email+ settings as needed.
3. Add any custom configurations for the app in AppConnect App Configurations.
4. Add any certificates that are required.
5. Choose a distribution option for the configuration and click Done.
6. The configuration is distributed to the subset of the devices to which the app is distributed.

Related topics

- See **Ivanti Email+ configuration field description (Ivanti Neurons for MDM)** for a description of the fields.
- See **Additional configurations using key-value pairs** for a complete list of custom key-value pairs.
- See the Certificates and S/MIME sections in **Additional configurations using key-value pairs**.

5.5. Ivanti Email+ installation on an iOS device (Ivanti EPMM and Ivanti Neurons for MDM)

Device users can install Email+ from a notification they receive on their iOS device, or from the app catalog on their device.

Ivanti Email+ for iOS installation from notification

After you send an installation request for Email+ for iOS, users receive a notification that prompts them to install the new or updated app. By tapping Install, Email+ for iOS is installed to the device.

Ivanti Email+ for iOS installation from the app catalog

When a featured app or an update to an installed app is published to device users, those users see a badge that appears on the corresponding tab in the app catalog. The number on the badge indicates the number of apps or updates available. The availability of an update is determined by comparing the version number for the installed app to that of the newly-published app.

After importing Email+ for iOS into the app distribution library, the app appears in Apps@Work on the device. Tap the entry for Email+ and follow the prompts to install the app.

5.6. Ivanti Email+ configuration field description (Ivanti Neurons for MDM)

To configure Email+ configuration on Ivanti Neurons for MDM, select the check box next to the configuration. The following table provides a description of the configuration fields for Email+ for iOS on Ivanti Neurons for MDM.

Item	Description
Email Address (Required)	Enter $.
Email Password	Enter the user's password for the ActiveSync server. If you provide a password, Email+ for iOS does not prompt the device user for the password. You can use the variable $
Exchange Host (Required)	Enter the fully qualified domain name of the ActiveSync server or the external hostname or IP address for Standalone Sentry.
Exchange Username (Required)	Enter $
SSL required	Check to secure communication to the ActiveSync server or Standalone Sentry using HTTPS. Select the check box unless you are working in a test environment.
Minimum Characters for GAL Search	Enter the minimum number of characters for Email+ for iOS to use for automatic Global Address List (GAL) lookup in Mail and Contacts. When the device user enters the specified number of characters of a particular name, Email+ for iOS searches the GAL and presents any matches to the device user. To enable GAL search, you must set the minimum number of characters for GAL search in your Microsoft Exchange server to the same value you set for this Email+ for iOS key. The default is 4.
App Identity Certificate	Select the App Identity Certificate created for Sentry. This field is required only if you are deploying Standalone Sentry that uses an identity certificate for device authentication.
Trust All Certificates	Check if you want Email+ for iOS to automatically accept untrusted certificates. Typically, you select the check box only if you are working in a test environment.
Prompt for Password Before Connecting to Server	Check if Email+ for iOS should prompt the user for the email password *before* attempting to connect to the email server. When it first launches and connects to the email server, Email+ for iOS provides the user's email password to the email server. If the field is unchecked, when Email+ for iOS first launches and connects to the email server, it does not provide the device user's email password to the server. After establishing a connection with the email server, Email+ for iOS prompts the user for an email password. If the email server limits the number of password attempts, it counts the connection as one failed attempt. Ivanti recommends checking this field if the email server allows only a small number of password attempts. For example, if the email server allows only three login attempts, setting this value to true means the device user gets three login attempts as specified by the email server.
Allow Safari Browser	Check to open links in Email+ in Safari. If the setting is checked, the values of email_url_scheme_http and email_url_scheme_https keys are ignored.
Allow Detailed Notifications (Required)	Check if you want Email+ for iOS to show the device user detailed notifications. The details can include sensitive information such as email subject, or event titles and times.

Item	Description
Show Pictures by Default	Check to enable the Show Pictures option. Device users automatically see images when opening an email. Device users can override the value you configure by toggling the Show Pictures option on or off. When changing the value of this key, Email+ does not change the Show Pictures option until after completing a full synchronization. A full synchronization occurs only when you change certain fundamental key-value pairs, like email_address, or when the device user uninstalls and reinstalls Email+ for iOS.
Allow Export Contacts	Check if you want to allow Email+ for iOS users to export Email+ for iOS contacts to an Email+ for iOS contacts group on the personal side of the device. Otherwise, enter false. When device users export Email+ contacts, device users can see the caller ID of incoming calls from phone numbers in the list of corporate contacts. Third-party apps can also access the corporate contacts. Contact details are exported, excluding the Notes field due to Apple limitations.
Limit Contact Export to Name and Number only	Check to limit export of Email+ contacts to only the name and number of the contacts. This option is available only if Allow Export Contacts is checked.
Allow Logging	Check if you want Email+ to log data to the device console, and allow the log file to be attached to a feedback email. This option is useful for problem diagnosis.
Default Email Signature	Enter the default email signature. The value of this key is the default email signature for all emails. However, the device user can define the default email signature at any time, overriding this value. After the user defines the default email signature, Email+ does not use the value, even if you update it.
Allow Send Feedback	Enter the email address to which app feedback is to be delivered. Use this key to send Email+ for iOS log messages to a particular email address.

6. Additional configurations using key-value pairs

The following describe how to customize Ivanti Email+ for iOS app behavior:

6.1. Key-value pairs for customization Ivanti Email+ for iOS

key-value pairs for customization describes the key-value pairs available to administrators to customize Email+ for iOS app behavior. These key-value pairs define app behavior such as providing detailed notifications to device users and export contacts from Email+.

> Key-value pairs marked as Ivanti EPMM only are not applicable to Ivanti Neurons for MDM. For Ivanti Neurons for MDM deployments, these key-value pairs are either provided as fields in Ivanti Neurons for MDM or are set automatically and do not require action from the administrator. See **Ivanti Email+ configuration field description (Ivanti Neurons for MDM)** for a description of the fields in Ivanti Neurons for MDM.

> Some values can use Ivanti EPMM variables, such as $EMAIL$. Ivanti EPMM substitutes the device user's value when sending the app configuration to the device.

You can configure and customize the following features with key-value pairs:

- **Required key-value pairs**
- **Background email check and user notifications**
- **Certificates**
- **S/MIME**
- **Manage contacts**
- **Syncing**
- **Maximum size for email**
- **Email attachments**
- **Open links in a browser**
- **Default signature**
- **SSL**
- **GAL search**

Key	Value: Enter/ Select one	Description
Required key-value pairs		
email_address (Ivanti EPMM only)	*Email address of the device user*	Typically, this field uses the Ivanti EPMM variable $EMAIL$. You can also use combinations of these Ivanti EPMM variables, depending on your ActiveSync server requirements: $USERID$, $USER_CUSTOM1$, $USER_CUSTOM2$, $USER_CUSTOM3$, $USER_CUSTOM4$.
email_device_id (Ivanti EPMM only)	$DEVICE_UUID_NO_DASHES$	Identifies the device to the ActiveSync server. Always use the Ivanti EPMM variable $DEVICE_UUID_NO_DASHES$.
email_exchange_host (Ivanti EPMM only)	*FQDN of the ActiveSync server or Standalone Sentry*	The fully qualified domain name of the ActiveSync server. If you are using a Standalone Sentry, enter the fully qualified domain name (FQDN) of Standalone Sentry. Example: mySentry.mycompany.com
email_exchange_username (Ivanti EPMM only)	*User ID for the ActiveSync server*	The user ID for the ActiveSync server. Typically, you use the Ivanti EPMM variable $USERID$. If your ActiveSync server requires a domain, use *<domain name>*\$USERID$. For example: mydomain\$USERID$. You can also use combinations of these Ivanti EPMM variables, depending on your ActiveSync server requirements: $EMAIL$, $USER_CUSTOM1$, $USER_CUSTOM2$, $USER_CUSTOM3$, $USER_CUSTOM4$. This KVP is case sensitive. Enter the key in lower case. Key value pair value behavior updated as part of MSAL changes: • The email_address KVP value is now is now used for authentication instead of email_exchange_username. • Microsoft Authentication Library (MSAL), which is now used instead of the deprecated Microsoft Azure Active Directory Authentication Library (ADAL). Thus, email_address and email_exchange_username should have same values for upgrade to Email+ 5.0. • After upgrading to Email+ 5.0.0, re-authentication is must to obtain a new token due to a limitation in the

Key	Value: Enter/ Select one	Description
		MSAL library. • Microsoft Authenticator will be used for Email+ authentication if it is installed.
Background email check and user notifications		
allow_detailed_notifications (Ivanti EPMM only)	• true • false	true: Device user sees detailed notifications. The details can include sensitive information such as email subject, or event titles and times. false: Notifications do not include any details. Default if key-value is not configured: false.
should_cache_tunneling_config	• true • false	Use the key should_cache_tunnelling_config along with the key allow_device_keychain. true: The configuration will be cached by AppConnect, as a result device user sees detailed push notifications and badge count (number of unread mails) after force closing the app. false: The configuration is not cached. Default if key-value is not configured: false.
Certificates		
allow_certificate_revocation_check	• true • false	true: Allows CRL check. Default if key-value is not configured: false
allow_device_keychain	• true • false	true: Email+ stores the decryption key received from the UEM client in the device keychain. This allows Email+ to access its credentials and check email when iOS launches it in the background, thus improving background email notifications. false: The AppConnect content decryption key is not stored on the device. Ivanti recommends that customers set this to true in conjunction with a strong device passcode. For more information see **Background email checks and user notifications**. Default if key-value is not configured: false
email_login_certificate (Ivanti EPMM only)	From the dropdown list	The device uses the certificate for authentication. See the Ivanti EPMM Device Management Guide for your deivce platform for information on configuring Certificate Enrollment settings. If the certificate is password-encoded, Ivanti EPMM automatically also sends another key,

Key	Value: Enter/ Select one	Description
		email_login_certificate_MI_CERT_PW, with the password as the certificate's value. This key is required if Sentry is configured to require certificates. Default if key-value is not configured: Certificates are not used.
email_trust_all_certificates (Ivanti EPMM only)	• true • false	true: Email+ automatically accepts untrusted certificates. Typically, you enter true only when working in a test environment. false: Email+ does not accept untrusted certificates. Default if key-value is not configured: false.
email_user_certificate_self_service (Ivanti EPMM only)	From the dropdown list	Allows the administrator to distribute certificates to device users. Users can then upload the certificates manually to the Ivanti EPMM user portal.
email_certificate_X where X is 1 through 10	From the dropdown list	You can designate up to ten certificate authority (CA) root certificates as trusted. Email+ imports the certificate into its keychain of trusted certificates, and trusts any certificates derived from the CA root certificate in its keychain. Designating a CA root certificate as trusted is necessary for the following: • You have configured device authentication in Standalone Sentry to require a certificate whose certificate authority is not a trusted CA. A common scenario for this case is if you are using a self-signed certificate or a certificate that is not derived from a well-known certificate authority. You specify this certificate to Email+ in the key email_login_certificate. It corresponds to the certificate you specified for device authentication in Standalone Sentry configuration in the Ivanti EPMM Admin Portal. • You have configured certificates for encrypting or signing S/MIME emails and these certificates are self-signed or not derived from a well-known certificate authority. You specify these certificates in the keys email_encryption_certificate and email_signing_certificate. Use .DER format instead of normal .PEM format for email_certificate_X certificates.
S/MIME		

Key	Value: Enter/ Select one	Description
email_encryption_certificate	From the dropdown list	Specifies the certificate to use for encrypting S/MIME emails. The UEM sends the contents of the certificate as the value. Email+ imports the key into the keystore and selects the certificate as the encryption certificate. If you change the certificate, Email+ imports the new certificate into the keychain and selects the new certificate as the encryption certificate. It leaves the previous certificate in the keychain. If you delete the key-value pair, Email+ leaves the certificate in the keychain, while changing its settings to specify that no certificate is selected as the encryption certificate. For more information about configuring S/MIME for Email+ for iOS, see **S/MIME support in Ivanti Email+ for iOS**. Default if key-value is not configured: Certificate is not configured. For S/MIME certificates use .DER format instead of normal .PEM format.
email_signing_certificate	From the dropdown list	Specifies the certificate to use for signing S/MIME emails. The UEM sends the contents of the certificate as the value. Email+ imports the key into the keychain and selects the certificate as the signing certificate. If you change the certificate, Email+ imports the new certificate into the keychain and selects the new certificate as the signing certificate. It leaves the previous certificate in the keychain. If you delete the key-value pair, Email+ leaves the certificate in the keychain and changes its settings to specify that no certificate is selected as the signing certificate. For more information about configuring S/MIME for Email+ for iOS, see **S/MIME support in Ivanti Email+ for iOS**. Default if key-value is not configured: Certificate is not configured. For S/MIME certificates use .DER format instead of normal .PEM format.
email_encryption_algorithm	• AES-256, • AES-192,	Configures encryption algorithim. The default value is set to SHA-256.

Key	Value: Enter/ Select one	Description
	• AES-128	
email_signing_digest	• SHA-1 • SHA-256 • SHA-384 • SHA-512	Configures signature algorithim. The default value is set to SHA-256. The restriction is empty by default. If there is no value or invalid value set, then SHA-1 is used.
S/MIME- Support for Retired Certs		
email_escrow_certificates	Each dictionary consists of the following two keys: • email_escrow_certificates • email_escrow_certificates_MI_CERT_PW	Use this option to use the multiple retired certificate for decrypting older messages. This value corresponding to this KVP is an array of dictionaries. • email_escrow_certificates: Is a base64 encoded p12 archive with certificate and private key. • email_escrow_certificates_MI_CERT_PW: Is a password string to unpack archives.
Manage contacts		
allow_export_contacts (Ivanti EPMM only)	• true • false	true: Allows Email+ users to export Email+ contacts to an Email+ contacts group on the personal side of the device. When device users export the contacts, they can see the caller ID of incoming calls from phone numbers in the list of corporate contacts. Third-party apps can also access the corporate contacts. false: Device users cannot export the Email+ contacts. They see the caller ID only for personal contacts. Default if key-value is not configured: false. Contact details are exported, excluding the Notes field due to Apple limitations.
limit_contact_export_to (Ivanti EPMM only)	• name_number • all	• name_number: limits the exported contact information to each contact's name, number, phonetic first name, and phonetic last name. • all: exports all contact information for each contact. This field is used only if allow_export_contacts is set to true.

Key	Value: Enter/ Select one	Description
		If you enter a value other than all or name_number, Email+ for iOS uses the value all. Default if key-value is not configured: all
email_safe_domains	*A comma-separated list of safe domains*	Ensure that there are no spaces before or after the comma. A wildcard in the domain name is supported. The only format supported for domain names with a wildcard is *.domainname.com. Entering * only will make all domains safe. Base domain is not included in the wildcard domain, it needs to be added explicitly if required. For example, *.domainname.com, domainname.com. Email addresses not in the safe domain list are displayed in red color in Email+. This configuration minimizes the risk that a user will accidentally send internal emails to external email addresses. You may want to use this key-value pair: • if your company policy requires this risk mitigation step. • if your company has multiple domains and you want to identify your company's domains as opposed to domains that are not your company domains. Example: mycompany.com,mycompany.net,internal.mycompany.com Default if key-value is not configured: Only the domain of the email account is safe.
email_alert_unsafe_domains	• true • false	true: Users see an alert if the recipients in an email or calendar invite include addresses that are not in a safe domain. For the alert to be displayed, the email_safe_domains key must also be configured. false: An alert is not displayed for addresses not in a safe domain. Default if key-value is not configured: false.
Syncing		
email_max_sync_period	• 0 • 1 • 2 • 3 • 4	Controls the maximum number of days for which emails are synced: • 0 = Download all emails. • 1 = Download emails received over the last day. • 2 = Download emails received over the last 3 days.

Key	Value: Enter/ Select one	Description
	• 5	• 3 = Download emails received over the last week. • 4 = Download emails received over the last 2 weeks. • 5 = Download emails received over the past month. Default if key-value is not configured: 0 Device users can change the interval to a value less than the default maximum. This feature is useful for regulatory purposes, if an organization requires device users to have no more than *n* days of emails on their devices. If the maximum email synchronization (email_max_sync_period) period is less than the default email synchronization period, then the maximum value is used.
email_default_sync_period	• 0 • 1 • 2 • 3 • 4 • 5	Controls the default time interval for which emails are downloaded: • 0 = Download all emails. • 1 = Download emails received over the last day. • 2 = Download emails received over the last 3 days. • 3 = Download emails received over the last week. • 4 = Download emails received over the last 2 weeks. • 5 = Download emails received over the past month. Default if key-value is not configured: 2 Ivanti does not recommend setting the value as 0, as downloading all emails could take a very long time, and take up too much space on the device.
Maximum size for email		
email_max_body_size	*A number*	Specifies the maximum size in megabytes permitted for each email that is received. This feature allows administrators to manage bandwidth and memory consumption on devices by restricting the maximum size of individual emails. If the size of the email is greater than the default or configured size, users are presented with the following message and the email cannot be downloaded: Email+ maximum message size exceeded. Default if key-value is not configured: 4 MB.
Email attachments		
email_max_attachment	*A number*	Specifies the maximum size in megabytes permitted for downloading each email or event attachment.

Key	Value: Enter/ Select one	Description
		If you set the maximum value to 10MB, a device user who receives an email that includes attachments of 3MB, 9MB, and 10MB will be able to download each attachment. If, however, a device user receives an email with an 11MB attachment, the following alert is displayed and users cannot download the attached file: Failed To Retrieve Attachment Email+ maximum attachment size exceeded. This limitation for outgoing emails is not controlled by this KVP., the following alert is presented: Warning: The message size exceeds 10 MB. Please confirm you would like to continue. Users have the option to either Cancel or Proceed. If users tap Proceed, the email is successfully sent. This feature allows administrators to manage bandwidth and memory consumption on devices by restricting the maximum size of individual email attachments. Default if key-value is not configured: 10 MB.
calendar_attachments	• true • false	Enabled viewing of files attached to calendar meeting invites.This feature requires Exchange Web Services to be configured. Email+ fetches calendar attachments via an EWS API. If Email+ is configured through Sentry, then additional key-value pair email_ews_host is needed with server address. The email_exchange_host is used automatically, but it is configured through Sentry email_ews_host. Default if key-value is not configured: false
MI_SHARED_GROUP_ID	*A unique, sufficiently complex alphanumeric string*	Required to enable attaching of files from Docs@Work. Ensure that the key-value pair is configured in the Docs@Work configuration as well and that the value is identical (including case) in both Email+ and Docs@Work configurations. The key is case sensitive. Enter the key in upper case. Configure mi_enable_doc_sharing with value true in the Docs@Work configuration.
MI_AC_ACCESS_CONTROL_ID	*A unique, sufficiently complex alphanumeric string*	Required to enable attaching of files from Docs@Work. Ensure that the key-value pair is configured in the Docs@Work configuration as well and that the value is identical (including case) in both Email+ and Docs@Work configurations. The key is case sensitive. Enter the key in upper case.

Key	Value: Enter/ Select one	Description
		Configure mi_enable_doc_sharing with value true in the Docs@Work configuration.
Open links in a browser Links in Email+ are opened by default in Web@Work. If Web@Work is not installed on the device, Email+ for iOS displays an error. However, administrators can specify the default browser to use when device users click links in Email+. Administrators can configure the default browser to be used for both HTTP and HTTPS links, using customized URL schemes. This allows finer control over the browser used to open HTTP and HTTPS links, respectively. Additionally, this key can be used to configure a customized browser as the one that launches when a device user clicks a link in Email+.		
allow_safari_browser (Ivanti EPMM only)	• true • false	true: Allows Email+ to open URLs (included, for example, in an email) in Safari. If the allow_safari_browser key is configured, the values of email_url_scheme_http and email_url_scheme_https are ignored. Default if key-value is not configured: false.
email_url_scheme_http	• mibrowser • googlechrome • firefox • microsoft-edge-http • touch-http	• mibrowser: Opens links in Web@Work for iOS • googlechrome: Opens links in Chrome. • firefox: Opens links in Firefox • microsoft-edge-http: Opens links in Microsoft Edge • touch-http: Opens links in Opera. Default if key-value is not configured: mibrowser
email_url_scheme_https	• mibrowsers • googlechromes • firefox	• mibrowsers: Opens links in Web@Work for iOS. • googlechromes: Opens links in Chrome. • firefox: Opens links in Firefox • microsoft-edge-https: Opens links in Microsoft Edge • touch-https: Opens links in Opera.

Key	Value: Enter/ Select one	Description
	• microsoft-edge-https • touch-https	Default if key-value is not configured: mibrowsers.
webatwork_install_link(Ivanti EPMM only. Not supported on Ivanti Neurons for MDM)	*URL for Web@Work*	If Web@Work is not installed on the device, device users are prompted to install Web@Work when they click on a webpage link in an email in Email+. If users accept the prompt, they are redirected to Apps@Work for installing Web@Work. The Web@Work URL is available in the app catalog in the Ivanti EPMM Admin Portal. In Ivanti EPMM, go to Apps > App Catalog, click on the Web@Work app, and then click Global. In the global settings, for App URL, click Copy Link to Clipboard. Paste the link as the value.
Default signature		
email_default_signature (Ivanti EPMM only)	*The default email signature*	The value of this key is the default email signature for all emails. However, the device user can define the default email signature at any time, overriding this key's value. After the user defines the default email signature, Email+ does not use the value in the key, even if you update it. Default if key-value is not configured: Sent by Email+ for iOS managed by MobileIron
SSL		
email_ssl_required (Ivanti EPMM only)	• true • false	true: Secures communication using https to the server specified in email_exchange_host. Default if key-value is not configured: true
GAL search		
gal_search_minimum_characters (Ivanti EPMM only)	*A number*	The minimum number of characters Email+ uses for automatic Global Address List (GAL) lookup in Mail and Contacts. When device users enter the specified number of characters of a name, Email+ searches the GAL and presents the matches that it finds. On your Exchange server, set the minimum number of characters for GAL search to the same value you set for this key. If you do not, GAL search will not work properly in Email+. Default if key-value is not configured: 4

Key	Value: Enter/ Select one	Description
gal_search_display_name	• true • false	true: Enables Display Name in Email+ Settings > Contacts by default. false: Disables Display Name in Email+ Settings > Contacts by default. Default if key-value is not configured: true
contacts_display_order	• first_last • last_first	Sets the default display order for contact names in search results. Device users can change the display order in Email+ in Settings > Contacts. The values are case sensitive; enter in lower case. first_last: Contact names in search results are displayed with first name followed by the last name. last_first: Contact names in search results are displayed with last name followed by the first name. Default if key-value is not configured: first_last.
Classification Markers		
email_security_classification_json	Is equal to JSON representation of JSON configuration. Is equal to JSON representation of classification configuration. For JSON. sample format. See, **Classification markers** section.	Allows the admin to configure Email Classification Markers, for secure sharing of Mail and Calendar events. The mail is marked with a marker that defines security of the mail. You can add any of the following markers to a mail: • Unofficial • Official • Secret • Protected • Top secret

6.2. Key-value pairs for customization Ivanti Email+ for iOS (Cont.)

You can configure and customize the following features with key-value pairs:

- Prompt the device user for password
- Keyboard extension
- Enable Show Pictures
- Photo library
- Calendar customization
- Notes customization
- Default network timeout
- App feedback
- Troubleshooting
- Miscellaneous

Table 1. key-value pairs for customization

Key	Value: Enter/ Select one	Description
Prompt the device user for password		
prompt_email_password (Ivanti EPMM only)	• true • false	true: Email+ prompts the user for the email password *before* attempting to connect to the email server. When Email+ first launches and connects to the email server, Email+ provides the user's email password to the email server. false: When Email+ first launches and connects to the email server, it does not provide the device user's email password to the server. After establishing a connection with the email server, Email+ prompts the user for an email password. If the email server limits the number of password attempts, it counts the connection as one failed attempt. Set the value of this key to true if the email server allows only a small number of password attempts. Example: It he email server allows only three attempts, setting this value to true ensures that device users get three attempts, not two attempts. Default if key-value is not configured: false
allow_prompt_password	• 0 • 1	0 = users are allowed access without a prompt for a password. 1 = users are prompted for a password to access email. Default if key-value is not configured: 1.
mail_password (Ivanti EPMM only)	*User's password for the ActiveSync server*	You can use the Ivanti EPMM variable $PASSWORD$ if you have checked Save User Password in Settings > Preferences. Ivanti EPMM then passes the user's password as the value to the device. If you plan to use the $PASSWORD$ variable, be sure to set Save User Password to Yes before any device users register. If a device user was registered before you set Save User Password, Email+ prompts the user to enter the password manually. Ivanti recommends deleting the key if the password is not being saved on Ivanti EPMM. Default if key-value is not configured: Email+ requests device users to enter the password.
Keyboard extension		
MI_AC_IOS_ALLOW_CUSTOM_KEYBOARDS	• true • false	true: Email+ allows the use of keyboards extensions. false: Email+ does not allow the use of keyboards extensions. This key-value pair is case sensitive. Default if key-value is not configured: false.
Enable Show Pictures		
show_pictures_default	• true	true: Enables the Show Pictures option. Device users automatically see images when opening an email.

Key	Value: Enter/ Select one	Description
(Ivanti EPMM only)	• false	false: Disables the Show Pictures option. Device users must tap Show Pictures to view images when opening an email. Device users can override the value you configure by toggling the Show Pictures option on or off. When changing the value of this key, Email+ does not change the Show Pictures option until after completing a full synchronization. A full synchronization occurs only when you change certain fundamental key-value pairs, like email_address, or when the device user uninstalls and reinstalls Email+ for iOS. Default if key-value is not configured: false.
Photo library		
allow_photo_library_access	• true • false	• true: Users can attach photos and video files from their personal photo library on the device. • false: Disables access to the personal photo library, including video files, from Email+. Device users cannot attach photos or videos from their personal photo library. However, users can take new photos or videos directly from the email they are composing in Email+ and attach to the email. This feature allows administrators to clearly separate work-related and personal content on device. Default if key-value is not configured: true.
Calendar customization		
calendar_default_reminder	• -1 • 0 • 5 • 10 • 15 • 30 • 60 • 120 • 1440 • 2880	Specifies the default calendar alert: • -1: No alert • 0: At the time of event • 5: 5 minutes before the event • 10: 10 minutes before the event • 15: 15 minutes before the event • 30: 30 minutes before the event • 60: 1 hour before the event • 120: 2 hours before the event • 1440: 1 day before the event • 2880: 2 days before the event Device users can edit the alert as desired after creating the event.

Key	Value: Enter/ Select one	Description
		Default if key-value is not configured: -1.
calendar_default_mode_tablet	• day • week • month • list	Sets the default Calendar view on an iPad. Device users can change the view in the Calendar's Settings. The device user's choice overrides the default set by the administrator. Default if key-value is not configured: week.
calendar_default_mode_phone	• day • month • list	Sets the default Calendar view on an iPhone. Device users can change the view in the Calendar's Settings. The device user's choice overrides the default set by the administrator. Default if key-value is not configured: day.
calendar_reset_view_threshold	*A number*	Sets the inactivity threshold after which the calendar view is reset to the default view. The inactivity threshold is measured in seconds. If the device screen is auto-locked or the app is in background for more than the configured time, the default view is loaded when users launch Calendar. Default if key-value is not configured: 300 seconds. The minimum value: 120 seconds
calendars_sync_period	The KVP can contain numbers 0, 1, 3, 6 where: 0 - sync all events 1 - sync events for 1 month (default) 3 - sync events for 3 months 6 - sync events for 6 months	Configures an extended sync period for calendar events. After the events are synced, the users can view and search events on all calendar screens.
Notes customization		
allow_notes_title	• true • false	true: Email+ users are presented with a separate title field to add a title to a note. false: A separate title field for a notes is not available, instead, the first line of the note is used as a title. Default if key-value is not configured: false.
Default network timeout		

Key	Value: Enter/ Select one	Description
default_network_timeout **App feedback**	*A positive integer*	Sets the app's default timeout for all ActiveSync network requests. The value is measured in seconds. Example: 30. In this example, for ActiveSync network requests, Email+ will timeout after 30 seconds. Default if key-value is not configured: 90 seconds
feedback_email_address (Ivanti EPMM only)	*An email address*	Device user app feedback and log messages are sent to the email address. Default if key-value is not configured: App feedback is not available to Email+ users.
Email watermark		
email_watermark email_watermark_parameters	Any alphanumeric string • textColor, • textSize, • horizontalSpacing, • verticalSpacing Where: textColor is #AARRGGBB and textSize, horizontalSpacing, and verticalSpacing are integer values.	Adds watermark in an email for mail view and mail compose screens supported. The text for the watermark is unique to a customer. Sets the watermark parameters such as text color, size, horizontal and vertical spacing.
Troubleshooting		
allow_logging	• true • false	true: Email+ logs data to the device console, and allows the log file to be attached to a feedback email. Entering true is useful for problem diagnosis. If an MSAL related error occurs during authentication, you can now share the Email+ logs using the "**Collect Logs**" button displayed in the error alert.

Key	Value: Enter/ Select one	Description
(Ivanti EPMM only)		Default if key-value is not configured: false.
allow_show_configuration	• true • false	true: Enables the display of configuration information while setting up Email+ on an iOS device. Set this value to true for test devices first, then disable the key value pair when it is time to roll out Email+ for iOS to device users. Default if key-value is not configured: false.
exit_on_configuration_error	• true • false	• true: Email+ simply shuts down without any notification if there is an error in the Email+ configuration that is pushed to the device. • false: If Email+ encounters an error in the configuration, device users are provided with the option to email the Email+ logs. The Email+ logs are helpful in debugging configuration errors. Not all configuration errors are considered critical. Example: A missing S/MIME signing or encryption certificate is not considered a critical error. Default if key-value is not configured: true
enable_calendar_dump	• true • false	• true: Enables calendar dump to Email+ feedback logs for troubleshooting. Calendar data is encrypted. • false: Disables calendar dump to Email+ feedback logs. Default if key-value is not configured: false
allow_logging_insecure	• true • false	true: Enables insecure logs for troubleshooting. false: Disables insecure logs for troubleshooting. Default if key-value is not configured: false.
allow_logging_responses	• true • false	true: Enables to record requests and responses for troubleshooting. false: Disables record requests and responses for troubleshooting. Default if key-value is not configured: false.
Miscellaneous		
disabled_features	• move_button • local_cache_mail • local_cache_calendar • local_cache_all	With the disabled_features key-value pair the admin can disable the values that are enabled by default: • **move_button**: Disables the Move button in the Move to screen in Email+. Emails are moved without confirmation when users tap on a folder. • **local_cache_mail**: Disables local caching of email addresses from mail (recipients from To, Cc, Bcc fields will not be added to local cache).

Key	Value: Enter/ Select one	Description
	• attach_files • openin_compose • document_viewer • email_priority_support • modern_mail_list • eas_16	• **local_cache_calendar**: Disables local caching of email addresses from calendar (recipients from Invitees field will not be added to local cache). • **local_cache_all**: Disables all local caching. • **attach_files**: Disables the Attach Files option in Email+. • **openin_compose**: Disables opening of files into Email+ from other apps. • **document_viewer**: Disables opening of attachments in Email+. Instead, users are provided the Open In ... option to choose an app in which to view the attachment. However, some attachments, such as text, .eml, audio, and certificate files are opened in Email+. The Email+ app does not support .msg file attachments in Open In option. Configure the value if you are also using the Watermark capability in Docs@Work. If you want watermarks to be shown on all documents, configure the value to disable the document viewer in Email+ and use Docs@Work exclusively. • **email_priority_support**: Disables the priority setting for emails. • **modern_mail_list**: Disables new mail list is, this feature is enabled by default. • **eas _16**: Disables support for Exchange ActiveSync16, for more information, see **ActiveSync16 for iOS devices**

enabled_features	• gmail_smart_folders
	• rms_support
	• fit_to_width
	• smime_suppress_certificate_email_check
	• calendar_delegation
	• personal_calendar
	• delegated_shared_mailbox
	• export_contacts
	• ai_tools
	• microsoft_teams_meetings
	• graph_api
		With the enabled_features key-value pair, the admins can enable the values that are disabled by default.
		• Enables smart folders, All mails, Spam, and Starred, for Gmail accounts.
		• Enables fetching, displaying and composing of the protected messages. If rms_support is enabled with configured server, list of fetched templates is displayed in Settings/Troubleshooting/Available Permissions. Due to ActiveSync protocol limitations, maximum of 20 templates are displayed in Email+.
		• Enables scaling down the size of an email to fit the width of the screen, but will not reduce it to less than 25% of its original size.
		• Enables automatic certificate verification using email address is suppressed and the user can manually add a certificate from GAL and Keystore using "Add Encryption Certificate" option.
		The **smime_suppress_certificate_email_check** feature works only when the **modern_sync_engine** value is configured for the **enabled_features** key-value pair.
		• Enables the delegated calendar
		• personal_calendar: To enable personal calendar function. This feature is enabled only for Modern Sync Engine which was enabled by default from Email+ 4.4.0 release onwards.
		• delegated_shared_mailbox: To enable the following options in the Settings section:
		  ◦ Delegated Mailbox
		  ◦ Delegated Calendar

Key	Value: Enter/ Select one	Description
		◦ Shared Mailbox • export_contacts: To automatically sync Email+ contacts to native contacts app. • ai_tools: To enable Apple Intelligence controls. • microsoft_teams_meetings: To enable microsoft teams in the Calendar settings. • graph_api: To enable Microsoft Graph-based functionality in Email+. When enabled, supported Microsoft 365 email and calendar features use Microsoft Graph API.
always_fetch_mime	• true • false	Use this option to set the default value for Email+ setting of **Always Fetch MIME**. Enable this option only when applications fail to launch when you click the links in the email body. Default if key-value is not configured: false
allow_callkit	• true • false	Use this option to disable CallKit functionality, which is restricted in China by Apple. Customers are advised to set allow_callkit to false for all their users in China. True - CallKit functionality enabled False - CallKit functionality disabled Default if key-value is not configured: true
disable_analytics	• true • false	Use this option to disable crash analytics. Default if key-value is not configured: false
filepass_key_identifier	A unique, sufficiently complex alphanumeric string.	This key allows admin to enable sharing documents securely between Docs@Work and Microsoft IntuneMAM protected Office365 apps through FilePass. The value for this key-value pair needs to be same for the all the supported productivity apps (Docs@Work, Email+, and FilePass) participating in File Sharing with Microsoft Office 365 apps.
migrate_from_email_exchange_host	The value is the previous email_exchange_host key that was set up on Ivanti EPMM.	This key allows the admin to configure Ivanti EPMM to Ivanti Neurons for MDM migration for the email exchange host. Post-migration of a device, the Email+ app needs to be re-synced to continue receiving and sending email messages. The content in local folders, such as Draft and Outbox, will be removed and no longer available. The Inbox folder should be refreshed for the email list to display.
eas_request_frequency_limit	The recommended value is 400.	This KVP is added to specify the throttling limit as X requests per 10 minutes, where X is the KVP value.
implicit_encryption	• true • false	The encryption logic for the Email+ 4.10.1 is now improved and enhanced. In case of any issues related to this update, it can now be rolled back by setting the **implicit_encryption = true** key-value pair in the AppConnect configuration
Microsoft Office 365		

Key	Value: Enter/ Select one	Description
authority and resource URL		
modern_auth_authority_url	https://login.microsoftonline.com/common	This KVP is added to specify Microsoft Office 365 authority url.
modern_auth_resource_url	https://outlook.office365.com	This KVP is added to specify Microsoft Office 365 resource url.
Phishing reporting		
report_phishing_address	email address	Enabling 'Report Phishing' option on View screen in the "More" menu. Phishing email will be sent to email address set in value.
Weekend alerts		
notifications_default_weekend_alerts	• true • false	The Email+ app now supports weekend alert notification for new users by default. When a user installs the app the first time or after the application is reinstalled after the app was wiped, the "Weekend Alerts" would be set to notifications_default_weekend_alerts value. (The reinstall of Email+ app will have no impact as Weekend Alerts settings are stored in KeyChain). The default is set to true
Mail organization		
organize_by_date	• true • false	Disables email threading for email messages. When a user installs the app for the first time or after the application wipe, the "**Organize By Thread**"is set to **organize_by_date** opposite value.
Avaya Workplace dialer app		
default_dialer	• avaya_workplace • native	When "avaya_workplace" value is selected, the Avaya Workplace app is used as a default dialer app.

Key	Value: Enter/ Select one	Description
		Native dialer is used as the default value, if **Avaya Workplace** dialer is not available or installed. In the Email+app **Settings > Contacts "Default dialer"** option is available with pre-selected default dialer.
allowed_dialers allow_universal_links **Custom Microsoft Authentication Library (MSAL) Configuration Options**	• avaya_workplace • true • false	The **allowed_dialers** kvp allows the Email+ app to use the **Avaya Workplace** dialer app as the default dialer along with the native dialer. This key allows the admin to open links in the associated application(like native iOS-Mail-App). If the value is set to false the link opens in the Web@Work app.
msal_resource_scope msal_client_id		Defines the resource scope required by MSAL to acquire access tokens for protected resources. Specify the multiple scopes with a single comma-separated string (for example, user.read, mail.read, and so on). This KVP is optional and required only if the customer has their own Email+ Azure application in their Azure tenant. Replaces the default application ID used by the Microsoft Authentication Library (MSAL) with a custom value. This KVP is optional and required only if the customer has their own Email+ Azure application in their Azure tenant.
msal_redirect_url		Redirects the url for MSAL This KVP is optional and required only if the customer has their own Email+ Azure application in their Azure tenant.
msal_internal_web_view msal_disable_authenticator_use	• true • false • true • false	true: MSAL should use the internal web view (wkWebView) false: MSAL should use Safari view controller Default if key-value is not configured: false. Modern Auth with certificate-based authentication is not supported by Safari view controller. This key allows the admin to disable the launching of Microsoft Authenticator to Authorization purpose. true: disables launching Microsoft Authenticator

Key	Value: Enter/ Select one	Description
		false: forces launching Microsoft Authenticator if it's installed Default if key-value is not configured: false. This KVP is optional; it's required for Per-App VPN configuration if Microsoft Authenticator is not tunnelled so it's not part of the authentication data flow.
Graph API Calls Configuration Options		
email_graph_host msal_graph_resource_scope msg_min_allowed_auth_mode realtime_notifications_protocol	graph.microsoft.com https://graph.microsoft.com/.default modern_auth 1	This key allows the admin to enable or disable Graph API operations.

6.3. S/MIME support in Ivanti Email+ for iOS

Ivanti Email+ for iOS includes support for Secure/Multipurpose Internet Mail Extensions (S/MIME). This functionality provides the following features:

- The device user sending the email can digitally sign the email.

On the receiving side, Email+ for iOS validates the sender's identity and determines whether the email has been tampered with.

- The device user sending the email can encrypt the email.

On the receiving side, Email+ for iOS decrypts the email.

- Email+ for iOS automatically encrypts emails when replying to or forwarding an encrypted email thread.

Using S/MIME requires a user certificate on the device running Email+ for iOS. You can import encryption certificates in one of two ways:

- **Pushing S/MIME certificates from Ivanti EPMM**

OR

- **Importing S/MIME certificates to the device through email**

Before you set up S/MIME for Email+ for iOS

Before you set up S/MIME do the following:

- Make users' public encryption keys accessible to all users.

To send an encrypted email, a user needs the recipient's public key. If you provide users' public keys in the Active Directory, Email+ for iOS uses global address lookup to retrieve a public key as needed.

Another way for one user to have the public key of another user is to receive an email from a user with one certificate for both signing and encryption. When receiving a signed email where the signing certificate and encryption certificate are the same, Email+ for iOS now has the sender's public key. The recipient can now send an encrypted email to the sender of the signed email.

- Make sure users' encryption certificates are the same on all devices.

Users need their private keys and certificates to read encrypted emails. A user's encryption key and certificate must be the same on all the user's email apps that use S/MIME, including desktop email apps.

- When an encryption key/certificate is renewed, the existing email on a device cannot be decrypted unless the original key certificate is available. Keep a backup copy of the encryption key and certificate or consider using a third-party escrow service.
- To restore an encryption key and certificate from backup, users can send themselves the key/certificate as an email attachment, as described in **Importing S/MIME certificates to the device through email**.

Pushing S/MIME certificates from Ivanti EPMM

Pushing S/MIME certificates from Ivanti EPMM is a two-step process:

1. **Enabling per-message S/MIME for iOS**
2. **Configuring key-value pairs**

Enabling per-message S/MIME for iOS

See the "Enabling per-message S/MIME for iOS" section in the Ivanti EPMM Device Management Guide for iOS and macOS Devices for iOS device to set up the encryption and signing certificates for S/MIME.

Configuring key-value pairs

The key-value pairs define the encryption and signing certificates to be used in Email+. The value for each key is the certificate enrollment setting you created. You enter the key-value pairs in the AppConnect app configuration you created for Email+ for iOS.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policy & Configs > Configurations.
2. Select the app configuration you created in **Creating an AppConnect app configuration for Ivanti Email+**.
3. Click Edit.
4. Add the following key-value pairs in the App-specific Configurations section:
    1. email_encryption_certificate: This key specifies the certificate to use for encrypting S/MIME emails. Select the SCEP setting you want to use from the dropdown list.
    2. email_signing_certificate: This key specifies the certificate to use for signing S/MIME emails. Select the SCEP setting you want to use from the dropdown list.

> Use of expired or revoked certificates for signing and encryption not supported. Also, the expired certificates are not displayed in the signing or encryption selection lists.

Pushing S/MIME certificates from Ivanti Neurons for MDM

To enable S/MIME encryption, set up the certificates you will use for S/MIME in Ivanti Neurons for MDM. You will reference the certificates in the Email+ configuration to distribute the certificates to devices. Certificates are sent to the devices to which the configuration is distributed. Email+ imports the certificates into the keychain and selects the certificates as the encryption and signing certificates, respectively. Device users can then use the certificates in Email+ for iOS.

Procedure

1. Set up certificates: Create a Certificate or Identity Certificate setting from Configurations > +Add. Before creating an Identity Certificate, you must have also added a certificate authority in Admin > Certificate Authority. See Ivanti Neurons for MDM Help for information about setting up certificates in Ivanti Neurons for MDM.
2. Configure the S/MIME key-value pairs in the Email+ configuration: The key-value pairs define the encryption and signing certificates to be used in Email+ for iOS. The value for each key is the certificate setting you created in **Set up certificates: Create a Certificate or Identity Certificate setting from Configurations > +Add. Before creating an Identity Certificate, you must have also added a certificate authority in Admin > Certificate Authority. See Ivanti Neurons for MDM Help for information about setting up certificates in Ivanti Neurons for MDM.**

Related topics

See the **key-value pairs for customization**, for the S/MIME key-value pairs for the encryption and signing certificates.

Importing S/MIME certificates to the device through email

Device users can import the signing and encryption certificates to their device from email.

Procedure

1. Device users email themselves the certificate they use for S/MIME as an attachment. The certificate must be sent as a PFX file.
2. Open the email using Email+ for iOS on the device
3. Tap to open the attachment. Email+ for iOS prompts the user for the certificate password.
4. Enter the certificate password. Email+ for iOS imports the certificate into its keychain.
5. Enable S/MIME signing and encryption in the mail settings in Email+ for iOS.
    1. In Email+ for iOS, tap Settings > Mail.
    2. Tap Security.
    3. Tap Sign. The user's signing certificate is automatically selected. Users may optionally tap Always Sign to always sign emails with their certificate, and Sign As Clear Text.
    4. Tap Encrypt. The user's signing certificate is automatically selected. Users may optionally tap Always Encrypt to encrypt every email they send through Email+ for iOS.

6.4. Background email checks and user notifications

Ivanti Email+ relies on iOS background execution to check for new email and to notify users. In the following cases Email+ may not be able to check for new email:

- To conserve battery power, iOS limits when third-party apps can run in the background. When Email+ is sent to the background, iOS occasionally allows Email+ to check for new email.
- iOS may terminate Email+ to reclaim memory for other apps. iOS may later decide to launch Email+ for iOS in the background to check email.

When Email+ is in the background and attempts to check email, as an AppConnect app that encrypts its content, Email+ must retrieve its encryption key from the UEM client. This requires an app flip to the UEM client, which cannot happen in the background. As a result, Email+ cannot retrieve the encryption key and therefore cannot verify working hours for notifications or check for new email.

To allow Email+ access to the encryption key even when it is in background, configure the allow_device_keychain key-value pair to allow Email+ to store the key in the device keychain. This allows Email+ to check for new email even when iOS launches it in the background.

> The allow_device_keychain key-value pair should only be used with a strong device passcode so as to secure the decryption key.

How Ivanti Email+ for iOS checks for new emails

The following describes how Email+ for iOS checks for new emails.

When the ...	Email+ for iOS...	iOS...
user launches Email+	checks for new email and notifies the user of new email	allows Email+ for iOS to run as usual
app is in the background	occasionally checks for new email and notifies the user, depending on iOS background refresh	might do a periodic background refresh

- iOS may sometimes terminate an app to preserve battery power or memory. If iOS terminates and then attempts to relaunch Email+ for iOS while the device is locked, then the decryption key is not accessible for reasons of security, and iOS cannot relaunch Email+ for iOS.
- iOS learns user habits and adjusts its background refresh parameters accordingly. As device users work with Email+ for iOS more frequently, iOS similarly launches Email+ for iOS in the background more frequently.

6.5. Configuring Ivanti Web@Work for iOS to open mailto links in Ivanti Email+ for iOS

Administrators can configure Web@Work for iOS to open mailto links in Email+ for iOS using key-value pairs. When device users click a mailto link in Web@Work for iOS, Email+ is automatically used.This feature allows administrators to maintain good security across the organization by ensuring that users go from a secure browser to a secure email application when clicking a mailto link.

Procedure

1. Select the Web@Work configuration in your UEM and click Edit.
2. In the Ivanti EPMM Admin Portal, go to Policies & Configs > Configurations.
3. In Ivanti Neurons for MDM, go to Apps.
4. For custom configurations, click Add to add a key-value pair.
5. Add the key mailto_prefix, and assign any of the following values:

Value	Value options	Description
email+app://<*Email+ for iOS app*>	email+app://email email+app://calendar email+app://contacts	Used for launching one of the apps within Email+ for iOS, such as the email app itself, calendar, or contacts.
email+launcher://<*browser URL scheme*>	email+launcher://mibrowser?url=mailto:	Sets Email+ as the default app to open mailto links.

5. Save the configuration.

6.6. Allow copy from Ivanti Email+ for iOS to other AppConnect apps only

You can configure data loss prevention policies (DLPs) for Email+ for iOS, which involves specifying whether the device user can copy content from Email+ for iOS to any other app. You can specify whether the device user can copy content from Email+ for iOS only to other AppConnect apps, rather than all other apps. This allows device users to share content without also allowing the content to flow to non-AppConnect apps.

Review your AppConnect for iOS DLP settings for copy/paste for Email+ for iOS. Ask these questions depending on your current setting:

Current setting	Ask yourself
Copy/Paste To is not allowed for Email+ for iOS.	• Would your security needs still be met if you allowed copying only to other AppConnect apps, but not to all other apps? • Does Email+ for iOS have content that a device user would want to copy to another AppConnect app? If you answered yes to these questions, change the Copy/Paste To setting for Email+ for iOS to allow copying only to other AppConnect apps.
Copy/Paste To is allowed for Email+ for iOS to all other apps.	• Would your security needs be better met by limiting copying only to other AppConnect apps, instead of all other apps? • Would the user feel limited if copying is allowed only to other AppConnect apps? If so, is the more limited user experience, but tighter content security the right trade-off for your needs? If you answered yes to these questions, change the Copy/Paste To setting for Email+ for iOS to allow copying only to other AppConnect apps.

Data loss prevention policies for Email+ for iOS are configured on the Ivanti EPMM Admin Portal in one of the following places:

- the AppConnect container policy for Email+ for iOS

See **Configuring the AppConnect container policy**

- the AppConnect global policy if you allow apps to be authorized without an AppConnect container policy and have no AppConnect container policy for Email+ for iOS.

See **Configuring the AppConnect global policy**

What Ivanti Email+ for iOS users see for copy/paste

When you limit copying from Email+ for iOS to only other AppConnect apps, the device user is able to copy from Email+ for iOS, but can paste only into the same app or other AppConnect apps. If the device user tries to paste into a non-AppConnect app, the content is not available.

The device user can also see the Copy/Paste To data loss protection policy setting in Mobile@Work at Settings > Secure Apps > Email+.

6.7. Email and calendar classification capabilities (Deprecated)

Classification capabilities provides the ability to manage protective markings to emails and calendar events. Email+ shows appropriate user interface fields to the user when viewing messages, replying to messages, or composing new messages, or creating new calendar events.

The messages that are sent through Email+, adds the markings to the subject line, header, and optionally on the top and the bottom of message body. Email+ supports two levels of markings:

- Classification - To identify the overall sensitivity of the message
- Distribution Limiting Markers - To limit the distribution

The different Classification and Distribution Limiting Markers (DLM) values are configured via key-value pairs as described in the following table;

Key-value pair	Description
email_classification_list	Enables the email classification feature. If present, it specifies the list of classification values to be used and all the supported permutations. The values are separated by a ";". For example: SEC=UNOFFICIAL;SEC=UNCLASSIFIED;DLM=For-Official-Use Only; DLM=Sensitive;DLM=Sensitive:LegalDLM=Sensitive:Personal;SEC=PROTECTED;SEC=PROTECTED,DLM=Sensitive;SEC=PROTECTED,DLM=Sensitive:Legal;
email_classification_version	Prefix the string to the x-header to represent the version of the standard supported. Default is "VER=2012.3,NS=gov.au". For example: X-Protective-Marking: VER=2012.3, NS=gov.au, SEC=UNCLASSIFIED, ORIGIN=neville.jones@ato.example.org
email_classification_default	Allows an administrator to specify the default classification value for all the new emails composed in Email+. If this value is not specified, then the user has to explicitly classify a message. The following value defaults to unofficial with no DLM SEC=UNOFFICIAL
email_classification_body	Enables whether the classification values are added to the message body (top and bottom) for emails that are sent (new emails as well as replies to existing emails). It is the Boolean value, if it is set to **True** then the message body is enclosed in Header and Footer. True/False. Default is True.
email_classification_lock_dlm	Enables the administrator to not allow any changes to the DLM value once it has been set in the original message. True/False. Default is False.
email_classification_required_string	A string value that is shown in the alert to the user asking the user to enter classification. Default is "Required".
email_classification_alert_string	A string value that is shown in the alert to the user asking the user to enter classification. Default is "Classification is required"
email_classification_security_label	A string value that is shown in the message compose user interface as label in the text field where the SEC value is selected.
email_classification_dlm_label	A string value that is shown in the message compose user interface as label in the text field where the DLM value is selected.

With Email+ 3.14.0 and newer the **Email and calendar classification capabilities** will be replaced with **Classification markers**. When you configure **email_security_classification_json** the older version of classification will be ignored.

6.8. Classification markers

Email classification markers allows secure sharing of Mail and Calendar events with internal and external audience. Configure Email classification marker using the **email_security_classification_json** key-value pair. To verify if classification is enabled on the Email+ app, go to **Settings > Troubleshooting > Email classification markers**.

The admin can configure and customize classification rules as required. When you compose a new mail using the Email+ app, select the **Classification** field to apply classification markers. The device users can set text value, color, alignment, and header or footer labels. You can assign colors to differentiate between field values for easy identification. When a mail with classification is saved as a draft or is stuck in the outbox. The classifications are also saved along with the mail message, you can edit the drafts mail to add or remove classification markers.

The Email classification markers consists of JSON formatted data. The following table describes the generic classification and sample format:

Classification	Value	Sample format

Fields	Array of JSON objects of the classification field configurations.
	• **name**: The name is a combination of alphabets and numbers. The first letter of the name should be in uppercase.
	• **title**: The text that is displayed to a user in classification picker. It consists of letters, numbers, and spaces. The default value is equal to **name** value. (Optional)
	• **description**: The text displayed in the Classification Picker, to provide users with more information about the Field. The default value is an empty string. (Optional)
	• **required**: When set to "true" a user is required to select a value for this field Boolean. The default value is set to false. (Optional)
	• **onReply**: Possible values:
	• **UPGRADE_ONLY** - allows to upgrade this classification field on email reply.
	• **LOCK** - prevents from changing this classification field on email reply. (Optional)
	• **ANY** - allows any change to this classification field on email reply. The default value is set to "ANY". (Optional)
	• **allowCustomValue**: When "true" a user would be able to add a custom value manually in the picker. Possible only when "onReply" is "ANY". Boolean. The default value is set to false. (Optional)
	• **selectionsMaxNumber**:When enabled, you can select multiple field values. The default value is set to 1.
		"fields":[
		{
		"name":"SEC",
		"type":"text",
		"title":"Security Classification",
		"description":"",
		"onReply":"UPGRADE_ONLY"
		},
		{
		"name":"ACCESS",
		"title":"Information Management Marker",
		"parent":"SEC",
		"required":false,
		"onReply":"LOCK"
		},
		{
		"name":"CAVEAT",
		"title":"Caveat",
		"required":false,
		"allowCustomValue":true
		}
		]

Values	Array of JSON objects of the classification fields values. • **$Fields.name$**: Array on the classification values for the appropriate field. The same field name can be declared multiple times for the unique "parentRange" • **value**: A value to be used for the specific classification marking. Must not contain semi-colons or be comprised of only spaces. Must be declared in the priority order, from the least secure to the most. (Required) • **title**: The text that is displayed as a classification value in the picker. The default value is equal to "value". (Optional) • **description**: The text that is displayed as a description to the specific classification value. The default value is an empty string. (Optional) • **defaultValue**: Classification selected by default on a message compose start. Object of the field-value pairs. The default is an empty field.	"values":[{ "SEC":[{ "value":"UNOFFICIAL", "title":"Unofficial", "description":"Non work-related email" }, { "value":"OFFICIAL", "title":"Official", "description":" Work-related emails that do not carry a security classification" }, { "value":"OFFICIAL:Sensitive",

Classification	Value	Sample format

	• **conditionality**: Defines the dependencies between different fields. One field may be dependent just to a single another one.
	• **target**: defines a parent field and its values in "field":["value_1", ...,"value_n"] format.
	• **dependent**: defines a child field with a list of values that a user would be shown when any of the "target" fields is selected.
		"title":"Official:Sensitive",
		"description":"Sensitive but not security classified information"
		},
		{
		"value":"PROTECTED",
		"title":"Protected"
		},
		{
		"value":"SECRET",
		"title":"Secret"
		},
		{
		"value":"TOP-SECRET",
		"title":"Top secret"
		},
		],
		
		},
		{
		"ACCESS":[{"value":"Personal-Privacy"},
		{"value":"Legal-Privilege"},
		{"value":"Legislative-Secrecy"}]
		},
		{
		"CAVEAT":[{"value":"SH:CABINET",
		title:"Cabinet"}, {"value":"REL:AU",
		title:"Australia"}],
		}
		],
		"defaultValue":{"SEC":"OFFICIAL",
		"ACCESS":"Personal-Privacy" },
		"conditionality":[
		{
		"targetField":"SEC",
		"dependentField":"ACCESS",
		"dependencies":[{"targetValues":
		["OFFICIAL","OFFICIAL:Sensitive","PROTECTED","SECRET","TOP-SECRET"],
		"dependentValues":["Personal-Privacy","Legal-Privilege","Legislative-Secrecy"]}]

Classification	Value	Sample format

		},
		{
		"targetField":"SEC",
		"dependentField":"CAVEAT",
		"dependencies":[{"targetValues":
		["PROTECTED","SECRET","TOP-SECRET"],"dependentValues":
		["SH:CABINET","$custom$"]
		}
		]
		},
		]
		}

Send Options	JSON object that defines the actions that would be applied to a message while sending and to a new event after creation.
	• **xHeader**: Object to define the X-Header key and value to be sent with each email.Optional.
	• **subjectSuffix**: Text to be appended to a classified message subject.
	• **bodyHeader** and **bodyFooter**: Formatted text to be appended to the start or the end of a message before an email sending. Optional.
	  ◦ **text**: text in Special formatting. **"\\n"** is resolved as a new line within the text.
	
	Required.
	  ▪ **color**: defines text color in #RRGGBB. Optional, default "#000000".
	  ▪ **alignment**: defines text alignment with "LEFT"|"CENTER"|"RIGHT" values. Optional, default "LEFT".
	  ▪ **style** - defines text style in HTML style format. When defined, "color" and "alignment" properties would be ignored.
	  ▪ **conditions**: allows to modify "bodyHeader" or "bodyFooter" values ("text", "color", "alignment") depending on "when" condition.
	  ▪ **when**: includes a field name and values. On an email sending with any of these classification values, a property from "result" would override the original "bodyHeader"/"bodyFooter" properties.
		"sendOptions": {
		"subjectSuffix" : "[(SEC=$SEC.value$){, }
		(ACCESS=$ACCESS.value$){, }
		(CAVEAT=$CAVEAT.value$){, }
		(CAVEAT=$CAVEAT.value_1$){, }
		(CAVEAT=$CAVEAT.value_2$)]",
		"bodyHeader":{
		"text":"($CAVEAT.title$: $CAVEAT.value$)
		{, }($CAVEAT.value_1$){, }
		($CAVEAT.value_2$)",
		"color":"#000000",
		"alignment":"CENTER",
		"conditions":[{"when":{"SEC":
		["PROTECTED"]},"result":
		{"color":"#FF0000"}}]
		},
		"bodyFooter":{
		"text":"($CAVEAT.title$: $CAVEAT.value$)
		{, }($CAVEAT.value_1$){, }
		($CAVEAT.value_2$)",
		"color":"#000000",
		"alignment":"CENTER",
		"conditions":[{"when":{"SEC":
		["PROTECTED"]},"result":
		{"color":"#FF0000"}}]
		},
		"allowNotClassified":true
		}

Classification	Value	Sample format
	• **allowNotClassified**: Boolean value that identifies if Email can be sent without classification selected. Boolean. Optional. Default value "false".	

ReceiveMarking	Defines a list of patterns to parse classification from a received message. The order in which the patterns are applied is defined in "priorities".
	• **xHeaderPatterns**: Rules to parse classification value from the X-Header.The X-Header key is searched by "headerName" values with left-to-right priority. (Optional)
	• **subjectSuffixPatterns**: Rules to parse classification value from an email subject.
	  ◦ **boundaries** define the range of characters in the subject to check for the classification. The "rules" are applied to the text starting from the first inclusion of "start" text and the last of "end". Whole subject is checked when not defined. (Optional)
	• **priorities**: Defines which patterns to fetch classification value. The default value is set to "priorities". ["xHeaderPatterns","subjectSuffixPatterns"]. (Optional)
		"receiveMarking": {
		"priorities": ["xHeaderPatterns","subjectSuffixPatterns"],
		"xHeaderPatterns":{ "headerName" : ["X-Protective-Marking", "X-Classification"],
		"rules":["SEC=SEC", "sec=SEC", "SEC:SEC", "sec:SEC", "ACCESS=$ACCESS$", "access=$ACCESS$", "ACCESS:$ACCESS$", "access:$ACCESS$", "CAVEAT=$CAVEAT$", "caveat=$CAVEAT$", "caveat:$CAVEAT$", "caveat:$CAVEAT$"],
		"separatorRegex":"[^a-zA-Z\d\-_\s]"
		},
		"subjectSuffixPatterns":{
		"rules":["SEC=SEC", "sec=SEC", "SEC:SEC", "sec:SEC", "ACCESS=$ACCESS$", "access=$ACCESS$", "ACCESS:$ACCESS$", "access:$ACCESS$", "CAVEAT=$CAVEAT$", "caveat=$CAVEAT$", "caveat:$CAVEAT$", "caveat:$CAVEAT$"],
		"boundaries":{"start":"[", "end":"]"},
		"separatorRegex":"[^a-zA-Z\\d\\-_\\s:]"
		}
		},

Version	**version** value should be defined to differ between JSON schemas. (Required)	classification version: 2.0.0

6.9. Crash reporting capabilities

Crashlytics is an enterprise grade service owned by Google and the Crashlytics SDK is embedded in productivity apps. Email+ implements the Crashlytics SDK. This enhancement helps to proactively address issues experienced by end-users.

Crashlytics related features:

- When an application crashes, the details of the crash are collected and uploaded to Crashlytics servers.
- The collected data includes OS version, RAM size, disk space, and so on, in addition to crash specific data.
- PII data is not uploaded to Crashlytics.
- Each device uses a unique random ID when uploading its crash reports to Crashlytics. This ID is also written to various application specific log files (such as Email+) and allows the engineering team to co-relate crash reports with specific customer provided log files.
- All data is maintained in the MobileIron Crashlytics account managed by the engineering team.
- Disable Crashlytics through the existing key-value pair **disable_analytics** for iOS Email+.

6.10. Allow logging on Ivanti Email+

Email+ logs data to the device console, and allows the log file to be attached to a feedback email.

For Ivanti EPMM, **allow_logging** key-value pair is used to configure logging.

For **Ivanti Neurons for MDM**, select the “allow Logging” check box in Email+ configuration to configure logging.

> The **allow_logging** KVP does not work on an AppConnect configuration.

6.11. Suppress certificate email check

The Email+ app can now suppress the verification of certificates using email address in case of email address or domain mismatch. To enable this feature add the **smime_suppress_certificate_email_check** value to the **enabled_features** key-value pair. When the Email+ app receives an email signed with a certificate, where the email address or domain differs from the sender's email address, the Email+ app associates the certificate with the sender's email address, so that the Email+ app can use the certificate when sending encrypted emails to the sender.

> The **smime_suppress_certificate_email_check** feature works only when the **modern_sync_engine** value is configured for the **enabled_features** key-value pair.

Adding possibility to search encryption certificates in Global Address List (GAL) and local Keystore. The Email+ app now supports manual searching and adding correct certificates from GAL and local Keystore and associating them with the correct recipient. The "Encryption certificates" section appears on the Contact Details screen with the list of the contact's associated certificates. Users can also add and delete certificates in the Contact Details screen.

The "Add encryption certificate" option allows to search the full name of the contact or email displaying the entries with encryption certificates. The Contact Details screen can be opened from Contacts as well as from the Compose Email screen and Event Details screen by tapping on the contact.

> If email in the S/MIME certificate doesn't correspond to email of the Exchange user, then this certificate cannot be returned by server when we search in GAL.

> The receiver should save the encryption certificate and make association for the certificate if the certificate in the email is different than the sender's email.

7. Real-time push notifications

These sections provide information on how to configure real-time push notifications. With real-time push notifications, notifications appear on the device as soon as a new email arrives on the Exchange server.

Push notifications at specified intervals

You can also set up push notifications at specified intervals (interval-based) as opposed to real-time push notifications. With interval-based push notifications the notification interval is configurable by the administrator. For information on how to configure interval-based push notifications for Ivanti EPMM deployments, see Cloud Notification Service for Email+ for iOS at **https://community.mobileiron.com/docs/DOC-4443**.

7.1. About real-time push notifications for Email+ for iOS

Email+ can be set up to receive real-time push notifications. Real-time notifications require additional setup with the Cloud notification service (CNS).

The Cloud notification service (CNS) is a cloud-based service hosted on Amazon Web Services (AWS) that provides real-time push notifications for Email+ for iOS users through Microsoft Exchange Web Services (EWS), Amazon Simple Notification Service (SNS), and Apple Push Notification Service (APNs).

- Need for a notification service
- How the notification service works

Need for a notification service

As a third-party app, Email+ for iOS is not permitted by iOS to execute for an unlimited period of time when the app is in the background. Only apps developed by Apple, such as the native mail app, are able to execute for an unlimited period of time in the background. Therefore, even though both native mail and Email+ use the ActiveSync protocol, only the native mail app can get real-time notifications.

The Cloud-based notification service (CNS) addresses this limitation by using the Apple APNS push notification service to notify users about new emails even when Email+ is running in the background on iOS devices. New emails also include calendar invites.

About the cloud notification service for Email+ for iOS

Cloud-based notification service enables Email+ for iOS to check new emails at regular intervals. The cloud server sends a periodic APNs message to devices with Email+ for iOS installed. The periodic APNs message triggers iOS to launch Email+ for iOS in the background, allowing Email+ to download new email from the mail server and notify device users accordingly. Administrators can optionally set the interval at which Email+ app checks for new email.

The notification service is configured on Ivanti EPMM using key-value pairs added to the AppConnect app configuration for Email+ for iOS.

Configuring the cloud notification service for Email+ for iOS involves the following main steps:

- Provide an email address to Ivanti for registration with the cloud notification service.

> No push notifications will be sent to this email address.

- Configure your organization ID and token (received from Ivanti) in Ivanti EPMM.

Before you begin

Configure, distribute, and install Email+ for iOS on all iOS devices.

Configuring cloud notification service for Email+

You must provide Ivanti with the name of your organization and an email address to which the cloud notification service is to be registered. Ivanti will register your organization with the service, and provide an organization ID and token (for security purposes) to be configured on Ivanti EPMM. The ID and token will allow Email+ for iOS to use the service.

To configure the cloud notification service for Email+ for iOS:

1. Navigate to the Customer Support Portal at **https://forums.ivanti.com/s/cloudnotificationservice**.
2. Enter your log in credentials.
3. Click the **Software** link at the top of the page.
4. On the Software page, click the **Cloud Notification Service (CNS) for iOS Email+** link.
5. To register for the service, click **Register Cloud Notification Service for iOS Email+**. Make sure popup windows are enabled in your browser.
6. On the registration page, enter the following information in the space provided:

- Email: The email address to which the cloud notification service is to be registered.
- Organization: The name of your organization. Make sure the email address you provide to Ivanti is an externally facing group address (not the email address of an individual), such as admin@example.com.

7. Click **Register**.
8. The registration page shows the email address and organization you entered, as well as an authorization token for the cloud server.
9. After Ivanti has registered your details, you will receive an email with the following information:

Item	Description
Name	The name of your organization as provided to Ivanti. For example, Acme Corp.
Email	The email address you provided to Ivanti, for example admin@example.com.
ID	An ID for your organization generated by Ivanti. For example: 000000b0-e000-00eb-a0ba-000000ba000c
Token	A randomly generated string from Ivanti, representing an authorization token for the cloud server.

10. Copy the token and paste it into a text editor.
11. In the Ivanti EPMM Admin Portal, go to **Policy & Configs > Configurations**.
12. Select the AppConnect app configuration you created for Email+.
13. Click **Edit**.
14. Add the following key-value pairs:

Key	Decription
notification_server_host	Enter the URL of the notification server: • https://cns-na1.mobileiron.com/PROD (interval-based push notifications) • https://cns.mobileiron.com/PROD (real-time notifications) The name of your organization as provided to Ivanti. For example, Acme Corp.
allow_device_keychain	Enter a value of true to enable Email+ for iOS to fetch email in the background.
notification_server_organization_id	Enter your organization's ID as provided to you by Ivanti. Use this KVP for interval-based push notifications.
notification_server_authorization	Copy and paste the token you received after you registered for the cloud notification service. Use this KVP for interval-based push notifications.
notification_interval	Optional. Enter the desired notification interval in seconds. The recommended interval is range is 5 minutes (300 seconds) to15 minutes (900 seconds). Use this KVP for interval-based push notifications.

15. Click **Save**. The updated AppConnect app configuration for Email+ for iOS will be sent to devices at the next sync interval.

Verifying the working of cloud notification service

To confirm if the cloud notification service is working:

Before you begin

Ensure that you have a registered email address to configure and access the Email+ app.

1. Install the Email+ iOS app and log in with the registered email address.
2. Place the Email+ app in the background mode without exiting the app.
3. Send a mail to the registered email address. New mail notification is received on the iOS device.

> The time taken to get new mail notification on your iOS device should be same as the interval at which the cloud service sends a push notification to iOS devices.

How the notification service works

Email+ uses Microsoft's Exchange Web Services (EWS) protocol to subscribe with Exchange to receive push notifications. As a result of the EWS subscription, Exchange sends a brief message to the Cloud-based notification service (CNS) when a new message is received. The Cloud notification service is hosted on Amazon Web Services (AWS) and uses Amazon's SNS service in conjunction with Apple's APNs service to send notifications to iOS devices. The APNs message triggers iOS to launch Email+ for iOS in the background, allowing Email+ to notify device users of new emails.

No sensitive user data or email content is transferred from Exchange to CNS. No corporate data or user identity information is stored on CNS, thus making the notification service safe and secure. Only the following information is sent from EWS to CNS:

CNS does not make any requests to the Exchange server.

The notification service is configured via the Unified endpoint management (UEM) platform using key-value pairs added to the AppConnect app configuration for Email+ for iOS.

1.	**Subscription workflow:**

a. Device registers with Apple APNs.

b. Devices registers with the EWS service on Exchange.

	2.	**A new email arrives on the Exchange server.**

	3.	**Exchange notifies Ivanti CNS.**

4.	**Ivanti CNS triggers APNs.**

5.	**APNs notifies the iOS device.**

	6.	**Notification workflow on Email+:**

> This feature requires users to be subscribed to CNS v2 for Real Time Notifications.

a. iOS displays a notification to the user indicating that there are new messages.

b. iOS wakes up Email+ in the background.

c. Email+ wakes up and fetches the email messages from the Exchange server via ActiveSync. Note that the email headers are fetched, and the body snippet is used for list view. The entire email body is not fetched.

d. Email+ replaces the previous notification with details of the new messages.

7.2. Standalone Sentry setup for real-time push notifications

The following sections provide the authentication to the Exchange and EWS service and the supported Standalone Sentry setup.

•	Exchange, real-time push notifications, and Standalone Sentry setup

•	EWS service and Standalone Sentry setup

Exchange, real-time push notifications, and Standalone Sentry setup

The following table shows the Standalone Sentry setup based on the required authentication and whether you are deploying real-time push notifications.

What is the authentication to Exchange?	Do you want real-time push notification?	Supported Standalone Sentry setup
Basic, NTLM	No	Enable ActiveSync on Standalone Sentry.
Basic	Yes	Enable ActiveSync and AppTunnel on Standalone Sentry. • Set up an AppTunnel service to tunnel Exchange Web Services (EWS). Device user experience: • Device users are prompted for user name and password for authentication to EWS.
Basic	Yes	Enable AppTunnel on Standalone Sentry.: • Set up an AppTunnel service for to tunnel Exchange Web Services (EWS) and Exchange ActiveSync (EAS) traffic. Device user experience: • Device users are prompted for user name and password for authentication to EWS and Exchange ActiveSync (EAS).
Basic Certificate	Yes No	Set up per app VPN with Tunnel.: Email+ must be an MDM managed app so that it can use Tunnel. Device user experience: • Device users are prompted for user name and password. Enable ActiveSync on Standalone Sentry.

Certificate, NTLM, Modern Auth	Yes	Setup per app VPN with Tunnel.
		• Email+ must be an MDM managed app so that it can use Tunnel.
		• If you are using certificates for authentication:
		  ◦ the certificate chain (root and intermediate) must be trusted by the Exchange server.
		  ◦ The certificate can be issued per user by a third-party CA or ADCS on Exchange.
		  ◦ The certificate is configured as a key-value pair in the Email+ configuration in the UEM.
		• For modern auth, Standalone Sentry setup is supported. The **PKCS-12** and **X.509** certificates should be configured on Ivanti EPMM.
		Modern auth uses ADFS that is exposed to the internet. The AppTunnel Sentry rules should be set to <ANY> **for adfs.company.com** on ports 443 and 49443.
		Device user experience:
		• Device users are not prompted for authentication.

EWS service and Standalone Sentry setup

The following table provides the supported authentication methods to the EWS service.

Setup	Basic Auth	Certificate Auth	NTLM Auth	Modern Auth
ActiveSync + AppTunnel	Yes Uses AppTunnel for EWS	No	No	No
AppTunnel only	Yes Uses AppTunnel for EWS and EAS.	No	No	No
Tunnel	Yes	Yes	Yes	Yes
No Sentry	Yes	Yes	Yes	Yes

7.3. Deployment use cases for real-time push notifications

This document addresses the following use cases:

- Email+ uses AppTunnel to tunnel EWS traffic to the Exchange server. This setup only supports basic authentication to the EWS service.
- Email+ uses Tunnel to tunnel all traffic to the Exchange server. This setup supports basic, modern auth, and identity certificates to authenticate to the EWS service.

> If your existing Email+ deployment uses a Standalone Sentry for ActiveSync and your Exchange Web Service (EWS) is set up to use certificates, you have to disable ActiveSync on Standalone Sentry and set up Tunnel.

7.4. Before you configure real-time push notifications

Before you configure real-time push notifications;

- Configure, distribute, and install Email+ for iOS.

Real time notification is supported for Email+ 2.4 0 for iOS through the most recently released version .

For information about installing Email+ for iOS, see **Configuring Ivanti Email+ for iOS**.

- For information about the EWS push notification service see Microsoft's documentation at **https://msdn.microsoft.com/en-us/library/office/dn458791%28v=exchg.150%29.aspx**
- Open port 443, for outbound only HTTPS requests, on your firewall to allow Exchange to send notifications to MobileIron CNS. The URL for the CNS server is https://cns.mobileiron.com/PROD. Alternately, you can enter the following IP addresses:
  - 13.56.49.23
  - 34.253.2.239

> Ivanti strongly recommends entering the URL for the notification server, as the IP addresses for the server might change.

- Set up your Exchange environment. See the following:
  - **For information about installing Email+ for iOS, see Configuring Ivanti Email+ for iOS.**.
  - **Configuring additional Exchange setup for identity certificates**.
- Ensure that Sectigoe is available in the Exchange trust store as a trusted certificate authority (CA). The Cloud notification service uses the Sectigoe CA.

Configuring EWS to send push notifications

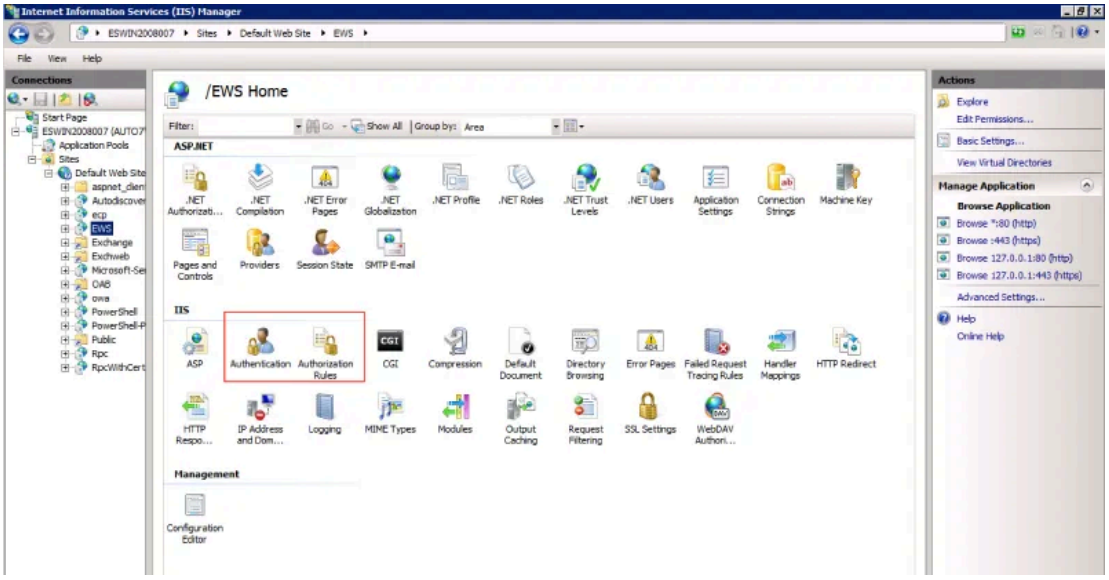
These steps are applicable for both Exchange 2010 and 2013 servers.

Before you begin

- You must have enabled EWS on the Exchange server.

Procedure

1. On the Exchange server, launch IIS Manager.
2. Go to Server > Sites > Default Web Site > EWS.
3. EWS home authentication and authorization



1. Verify that the Authentication and Authorization Rules roles are added to IIS.
2. Open Authentication and Enable Basic Authentication.
3. Open Authorization Rules and add rule to Allow for All users if it was not added automatically.

Next steps

- If your setup uses Tunnel and identity certificates to authenticate with EWS and ActiveSync, do the additional setup on the Exchange server described in **Configuring additional Exchange setup for identity certificates**.
- Once you have set up your Exchange environment, go to **Overview of configuration on Ivanti EPMM**.

Configuring additional Exchange setup for identity certificates

Perform these steps only if your setup uses Tunnel and identity certificates for authentication to EWS and ActiveSync.

Procedure

1. On the Exchange server, launch IIS Manager.
2. Go to Server > Sites > Default Web Site > EWS.
3. Click on SSL Settings.
4. Check Require SSL.
5. For Client certificate, select Accept.
6. In the EWS directory, click on Configuration Editor and browse to the clientCertificateMappingAuth option. Set the value for the option to True.
7. In the EWS directory, click on Authentication and enable the Windows Authentication option. Disable all other authentication types.

Next steps

- Once you have set up your Exchange environment, go to **Overview of configuration on Ivanti EPMM**.

7.5. Overview of configuration on Ivanti EPMM

This section provides an overview of the steps required to set up Email+ for real-time push notifications on Ivanti EPMM. Depending on your authentication requirements, use one of the following setup to tunnel Exchange Web Services (EWS) traffic:

- Using Tunnel to tunnel EWS traffic (Ivanti EPMM)

OR

- Using AppTunnel to tunnel EWS traffic (Ivanti EPMM)

Using Tunnel to tunnel EWS traffic (Ivanti EPMM)

This section provides the main steps for configuring real-time notifications with Email+ for iOS on Ivanti EPMM if you are using Tunnel to tunnel EWS traffic.

Before you begin

- Complete the setup described in Before you configure real-time push notifications.

Procedure

	1. Set up Tunnel.

> Email+ must be an MDM managed app so that it can use Tunnel.

	2. If your EWS setup uses either NTLM, modern auth, or identity certificates for authenticating to the EcWS service, create a SCEP certificate enrollment setting. Skip this step if your EWS setup uses basic authentication.

3. Update the Email+ AppConnect app configuration.

Related topics

- See Ivanti Tunnel for iOS Guide to set up Tunnel on Ivanti EPMM.
  - Configuring SCEP settings.
  - Updating the AppConnect app configuration for Ivanti Email+

Using AppTunnel to tunnel EWS traffic (Ivanti EPMM)

This section provides the main steps for configuring real-time notifications with Email+ for iOS on Ivanti EPMM if you are using AppTunnel to tunnel EWS traffic.

Before you begin

- Complete the setup described in Before you configure real-time push notifications.

Procedure

1. Add an <ANY> AppTunnel service in Standalone Sentry settings.
2. Update the Email+ AppConnect app configuration.

Related topics

- Configuring an AppTunnel service.
- Updating the AppConnect app configuration for Ivanti Email+.

7.6. Description of configurations in Ivanti EPMM

This section provides a more detailed description of the configuration steps referenced in **Overview of configuration on Ivanti EPMM**. The following configurations are described:

- **Configuring SCEP settings**
- **Configuring an AppTunnel service**
- **Updating the AppConnect app configuration for Ivanti Email+**

Configuring SCEP settings

Create a SCEP setting if your Exchange server and the EWS service require certificate authentication. You will reference the name of SCEP setting in the AppConnect configuration for Email+ to generate the login certificate for Email+, so that the Exchange server and EWS trust the device.

Procedure

1. In the Admin Portal, go to Policies & Configs > Configurations.
2. Select Add New > Certificate Enrollment > SCEP.
3. In the New SCEP Setting window, configure the settings based on your SCEP requirements.
4. Click **Save** to save the SCEP setting.
5. Click **OK** to dismiss the prompt indicating the successful creation of your SCEP setting.
6. You will reference this SCEP setting in the AppConnect app configuration for Email+ using the key email_login_certificate.

Related topics

- "Configuring SCEP" in the Ivanti EPMM Device Management Guide for iOS and macOS Devices.

Configuring an AppTunnel service

You create an AppTunnel service in Standalone Sentry as part of an AppTunnel setup.

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 Ivanti Sentry Installation Guide.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Services > Sentry.
2. Edit the entry for the Standalone Sentry that supports AppTunnel.
3. In the App Tunneling Configuration section, under Services, click + to add a new service.
4. Use the following guidelines to configure an AppTunnel service:

Item	Description
Service Name	Select <ANY>. The Service Name is used in the AppConnect app configuration for Email+.
Server List	Select the Standalone Sentry
TLS Enabled	NA
Proxy/ATC	NA
Server SPN List	NA

5. Click Save.

Updating the AppConnect app configuration for Ivanti Email+

Update the AppConnect app configuration for Email+ for iOS, so that Email+ on iOS devices is authorized to get real-time notifications from CNS.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policy & Configs > Configurations.
2. Select the AppConnect app configuration you created for Email+.
3. Click Edit.
4. Add an AppTunnel rule that points to the Standalone Sentry on which you configured the AppTunnel service.
    1. For URL Wildcard, enter the Exchange server's IP address or FQDN.
5. For Identity Certificate, select the Certificate Enrollment setting you configured for Standalone Sentry. You would have created the Certificate Enrollment setting as part of the Standalone Sentry setup for identity certificate with Pass through.
6. Add the necessary key-value pairs.
7. Click Save.
8. Ensure that the configuration is applied to the labels that contain the devices to which you want to push the configuration. The updated AppConnect app configuration for Email+ for iOS will be sent to devices at the next sync interval.

Related topics

See **Key-value pairs for real-time push notifications** for a list of key-value pairs.

7.7. Overview of configuration on Ivanti Neurons for MDM

This section provides an overview of the steps required to set up Email+ for real-time push notifications on Ivanti Neurons for MDM. Depending on your authentication requirements, use one of the following setup to tunnel Exchange Web Services (EWS) traffic:

- Using Tunnel to tunnel EWS traffic (Ivanti Neurons for MDM)

OR

- Using AppTunnel to tunnel EWS traffic (Ivanti Neurons for MDM)

Using Tunnel to tunnel EWS traffic (Ivanti Neurons for MDM)

This section provides the main steps for configuring real-time notifications with Email+ for iOS on Ivanti Neurons for MDM if you are using Tunnel to tunnel EWS traffic.

Before you begin

- Complete the setup described in Before you configure real-time push notifications.

Procedure

	1. Set up Tunnel.

See Ivanti Tunnel for iOS Guide to set up Tunnel on Ivanti Neurons for MDM.

> Email+ must be an MDM managed app so that it can use Tunnel.

	2. If your EWS setup uses either NTLM, modern auth or identity certificates for authenticating to the EWS service, create a SCEP certificate enrollment setting. Skip this step if your EWS setup uses basic authentication.

See Configuring Identity certificate setting.

	3. Update the Email+ app configuration.

See Updating the app configuration for Ivanti Email+.

Using AppTunnel to tunnel EWS traffic (Ivanti Neurons for MDM)

This section provides the main steps for configuring real-time notifications with Email+ for iOS on Ivanti Neurons for MDM if you are using AppTunnel to tunnel EWS traffic.

Before you begin

- Complete the setup described in Before you configure real-time push notifications.

Procedure

1.	Add a custom HTTP service to the Standalone Sentry profile.

See Configuring a custom HTTP service.

	2.	Update the Email+ app configuration.

See Updating the app configuration for Ivanti Email+.

7.8. Description of configurations in Ivanti Neurons for MDM

This section provides a more detailed description of the configuration steps referenced in **Overview of configuration on Ivanti Neurons for MDM**. The following configurations are described:

- **Configuring a custom HTTP service**
- **Configuring Identity certificate setting**
- **Updating the app configuration for Ivanti Email+**

Configuring a custom HTTP service

You create an AppTunnel service in Standalone Sentry as part of the AppTunnel setup.

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 Ivanti Sentry Guide for Ivanti Neurons for MDM.

Procedure

1.	**In Ivanti Neurons for MDM, go to Admin > Sentry.**
2.	**Edit the entry for the Standalone Sentry profile that supports AppTunnel.**
3.	**In Services, click Custom HTTP to add a new service.**
4.	**Use the following guidelines to configure the service:**

Item	Description
Service Name	Enter a name to identify the service. The Service Name is used in the Email+ app configuration.
Server Authentication	Select Pass through (Basic Authentication).
All destinations (forward proxy)	Selected by default.

5.	**Click Save.**

Related topics

See "Configuring Standalone Sentry for app tunneling" in the Ivanti Sentry Guide for Ivanti Neurons for MDM for more information on creating an AppTunnel service.

Configuring Identity certificate setting

You need to create the SCEP setting if your Exchange server and the EWS service require certificate authentication. You will reference the name of SCEP setting in the AppConnect configuration for Email+ to generate the login certificate for Email+, so that the Exchange server and EWS trust the device.

Before you begin

Create a certificate authority in Admin > Certificate Authority.

Procedure

1.	**In Ivanti Neurons for MDM, go to Configurations.**
2.	**Click Add > Identity Certificate.**
3.	**Fill in the following fields for the certificate configuration:**

- Name: Enter brief text that identifies this certificate setting.
- Description: Enter additional text that clarifies the purpose of this SCEP setting.
- Certificate Distribution: Select Dynamically Generated.
- Source: Select the Certificate Authority you created in Admin > Certificate Authority.
- Subject: CN=$
- Key size: 2048

4.	**Test the configuration, and click Next.**
5.	**Click Done to save the configuration.**

You will reference the certificate configuration in the app configuration for Email+ using the key email_login_certificate.

Updating the app configuration for Ivanti Email+

Update the app configuration for Email+ for iOS with key-value pairs, so that Email+ on iOS devices is authorized to get real-time notifications from CNS.

Procedure

1.	**In Ivanti Neurons for MDM, go to Apps > App Catalog.**
2.	**In the App Catalog, click on Email+.**
3.	**Click App Configurations.**
4.	**In App Configurations Summary, click on AppTunnel to add an AppTunnel rule.**

Enter a name for the configuration.

Select the Sentry profile in which you configured the custom HTTP service.

Select the custom HTTP service you created for real-time push notifications in the Sentry configuration.

For URL Wildcard, enter the Exchange server's IP address or FQDN.

Select the distribution for this configuration.

Click Save.

	5.	**In App Configurations Summary, click on Email+ Configuration to add the necessary key-value pairs in AppConnect Custom Configuration.**

	6.	**Select the app distribution.**

	7.	**Click Update.**

Related topics

See **Key-value pairs for real-time push notifications** for a list of key-value pairs.

7.9. Keys for real-time and interval-based push notifications (Ivanti EPMM and Ivanti Neurons for MDM)

- Key-value pairs for real-time push notifications
- Key-value pairs for push notifications (interval-based)

Key-value pairs for real-time push notifications

The following keys are applicable to configuring real-time push notifications:

- notification_server_host

For real-time push notifications enter the following value: cns.mobileiron.com/PROD.

- allow_realtime_notifications
- email_ews_host
- eas_min_allowed_auth_mode
- ews_min_allowed_auth_mode
- notification_resubscription_interval
- allow_device_keychain
- subscription_valid_until
- sentry_server_host
- should_cache_tunneling_config

The following table describes the key-value pairs applicable for real-time push notifications.

Key	Value: Enter/Select one	Description
allow_realtime_notifications	true	Enables real-time push notifications.
notification_server_host	*The URL of the notification server*	The URL for the notification server for real-time push notifications is cns.mobileiron.com/PROD. Alternately, you can enter the following IP addresses: 13.56.49.23 34.253.2.239 Ivanti strongly recommends entering the URL for the notification server, as the IP addresses for the server might change.
email_ews_host	*Exchange server address* for the EWS host	Explicitly sets the EWS host address for real-time push notifications, as opposed to the value configured for email_exchange_host. Enter the IP address or DNS of the EWS host. The DNS name must be in the following format: case2010.xyz.com. Do not prepend https or full path name. For Office 365, enter outlook.office365.com. This key-value pair is required if your Standalone Sentry is the email host, i.e. the email_exchange_host key points to the Standalone Sentry FQDN.
eas_min_allowed_auth_mode	• basic • ntlm • cert_base • modern_auth	Defines the authentication method to the Exchange ActiveSync service. • basic: Select if you are using Basic authentication (user name and password) • ntlm: Select if you are using NTLM authentication • cert_base: Select if you are using identity certificates for authentication • modern-auth: Select to enable modern auth for corresponding protocol If a key-value pair is not configured, the default authentication method is Basic. If you have configured ntlm or cert_base or modern_auth, and there are errors in your configuration, the authentication method defaults to basic.
ews_min_allowed_auth_mode	• basic • ntlm • cert_base • modern_auth	Defines the authentication method to the Exchange EWS service. • basic: EWS uses basic authentication (User name and password) • ntlm: EWS uses NTLM authentication • cert_base: EWS uses identity certificates for authentication.

Key	Value: Enter/Select one	Description
		For this value, Microsoft Office 365 is not supported. • modern_auth: user authentication on first application launch or when token cannot be refreshed or renewed If you have configured ntlm or cert_base or modern_auth, and there are errors in your configuration, the authentication method defaults to basic. Default value if no key-value is configured is basic.
notification_resubscription_interval	*A number*	Optional. Sets the interval when Email+ resubscribes to receive real-time push notifications. The resubscription interval is in minutes. If a key-value pair is not configured, the default resubscription interval is 60 minutes.
subscription_valid_until	The default value for "validUntil" is 1440 minutes	Sets the duration of the subscription.
sentry_server_host	host name address	Installs Sentry, to support VIP notification. Once Sentry is installed, the "VIP Allowed After Work Hours" option is enabled.
should_cache_tunneling_config Add the following key-value pairs if you are using an identity certificate for authentication	• true • false	Use the should_cache_tunnelling_config key along with the allow_device_keychain. true key. The configuration is cached by AppConnect, as a result device user can view detailed push notifications and badge count (number of unread mails) after force closing the app. false key: The configuration is not cached. Default if key-value is not configured: false.
email_exchange_username	$USERID$	The user ID for the ActiveSync server.
email_exchange_host	*<exchange_real_address>*	The fully qualified domain name of the ActiveSync server.
email_trust_all_certificates	true	Email+ automatically accepts all certificates.
email_login_certificate	*<name of the identity*	**Ivanti EPMM:** Name of the SCEP setting in Ivanti EPMM. **Ivanti Neurons for MDM:** Name of the Identity certificate configuration in Ivanti Neurons for MDM

Key	Value: Enter/Select one	Description
	certificate configuration>	
email_ssl_required allow_logging	true true	Secures communication using https to the server that you specified in email_exchange_host. Email+ logs data to the device console, and allows the log file to be attached to a feedback email.
email_device_id	$DEVICE_UUID_NO_DASHES$	Identifies the device to the ActiveSync server.
email_address	$EMAIL$	Email address of the device user.
feedback_email_address Add the following key-value pairs if your deployment includes Email+ versions 2.3.4 and less and the devices require interval-based push notifications: • notification_server_organization_id • notification_server_authorization For a description of the key-value pairs see **Key-value pairs for push notifications (interval-based)**. Email+ versions 2.3.4 and less do not get real-time notifications.	*An email address*	Device user app feedback and log messages are sent to the email address.

Key-value pairs for push notifications (interval-based)

The following keys are applicable to configuring push notifications:

- notification_server_host

For push notifications enter the following value: cns-na1.mobileiron.com/PROD.

- notification_server_organization_id
- notification_server_authorization
- notification_interval
- allow_device_keychain

The following table describes the key-value pairs applicable for interval-based push notifications.

Key	Value: Enter/Select one	Description
notification_server_host	*The URL of the notification server*	The URL for the notification server for real-time push notifications is cns-na1.mobileiron.com/PROD.
notification_interval	*A number*	The desired notification interval in seconds. The recommended interval is 15 minutes, or 900 seconds. The minimum interval is 5 minutes, or 300 seconds. This key is ignored if real-time push notifications is configured. Default value if no key-value is configured: 900 seconds.
allow_device_keychain	true	Enables Email+ to fetch email in the background.
notification_server_organization_id	*ID provided by Ivanti*	Organization ID provided by Ivanti.
notification_server_authorization	*Token provided by Ivanti*	Token for the cloud notification service.

7.10. Verifying that the Ivanti Neurons for MDM notification service is working

After configuring real-time push notification, verify that the service is working.

Procedure

1. **Obtain a test iOS device with an email address you can access configured on it.**
2. **Ensure that Email+ for iOS is installed to the device.**
3. **In Email+, go to Settings > Notifications and verify that your device is subscribed with EWS.**

The following message appears in the Mail Alerts section:

You are subscribed to real-time push notifications.

4. **Place the Email+ app in the background without exiting the app.**
5. **From your desktop, send an email to yourself, using the email address configured on the test iOS device.**
6. **Watch for a new mail notification from Email+ for iOS on the test device.**

8. Troubleshooting Ivanti Email+ for iOS

The following describe some tools for troubleshooting Email+ for iOS:

8.1. Setting up logging for Email+ for iOS (Ivanti EPMM)

You can troubleshoot user issues with Email+ for iOS by collecting logs and sending them to an email address you can access. You then ask the device user to reproduce the issue so that you can view logging data. You can also diagnose your configuration before rolling out Email+ for iOS to device users.

Procedure

1.	**In the Ivanti EPMM Admin Portal, go to Policy & Configs > Configurations and select the AppConnect configuration you created for Email+ for iOS.**

2.	**Click Edit.**

3.	**In the App-specific Configurations section, enter the key allow_logging with value as true.**

This allows Email+ for iOS to log data to the device console.

4.	**Enter the key feedback_email_address with value as a valid email address which you can access.**

Email+ for iOS sends the collected log data to the email address entered here.

5.	**If you are diagnosing a configuration, enter the key allow_show_configuration with value as true.**

When set to true, Email+ for iOS shows all configured key-value pairs for diagnostic purposes. Disable this setting after diagnosis is complete.

6.	**Click Save.**

7.	**Force a check-in on the user's device to ensure the modified AppConnect app configuration for Email+ for iOS is sent to that device:**

a. Go to Users & Devices > Devices.

b. Select the checkbox for the device.

c. Click Actions > Force Device Check-in.

The Force Device Check-In dialog appears.

d. In the dialog, confirm the user and device information and enter a note.

e. Click Force Device Check-in.

The device user will now see a Feedback icon in Email+ for iOS.

	8.	**Ask the device user to reproduce the problematic action and tap the Feedback button.**

Email+ for iOS log data will be collected and emailed to the address you provided.

Related topics

See **Key-value pairs for customization Ivanti Email+ for iOS** for additional key-value pairs for troubleshooting.

8.2. Detailed logging for AppConnect apps for iOS (Ivanti EPMM)

For more information about logging for AppConnect apps for iOS, see the section “Detailed logging for AppConnect apps for iOS” in the AppConnect and AppTunnel Guide.

Ivanti Email+ crash recovery

Email+ has a built-in crash recovery mechanism that is triggered in the event that Email+ consistently crashes upon launch. If Email+ crashes three times consecutively within a short interval each time it is launched, the consecutive crashes are considered a catastrophic failure that is associated with some internal data. In this case, the app reconfigures and the reconfiguration wipes all cached data.

9. What users see

The user can manage notifications for the Ivanti Email+ app.

The following sections describe how you can use the notifications option on Email+ for iOS:

9.1. Real-time push notifications

Real-time push notifications allow Email+ users to receive notifications about new emails as soon as the emails arrive in the Inbox. Previous versions of Email+ supported periodic notifications for new emails at an interval set by the IT department. The following sections address some questions you may have about real-time notifications:

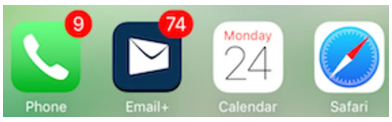
- How will I receive Email+ notifications?
- How do I change the notification settings?
- Why do I see two notifications for each email?
- Why am I not receiving Email+ Notifications?
- How do I turn on/off notification details on the lock screen?

How will I receive Email+ notifications?

Email+ notifications are displayed on your device as:

- A new email notification in the Notification Center on your device.
    - Badging of the Email+ app icon on the Home Screen.

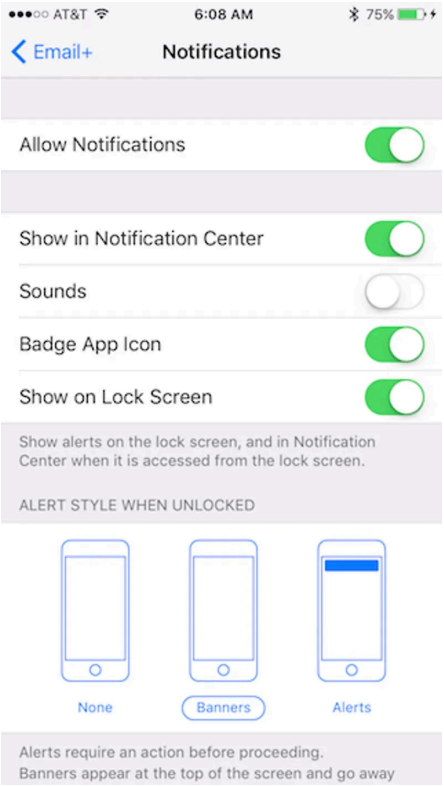
Figure 1. Email+ app icon badging



How do I change the notification settings?

To change the notifications settings for Email+, in iOS device Settings, tap Email+ > Notifications.

Figure 2. Email+ notifications options in device settings



Why do I see two notifications for each email?

If real-time notifications are enabled, the Email+ app displays two notifications for each new email. The first notification is sent by Apple APNs and shows up immediately on the lock screen (depending on the Emai+ Notification settings in your device Settings). This notification has the text: "You have new messages". The second, more detailed, notification is sent by the Email+ app if the app is running in the background. The Email+ app fetches the email summary for the new unread email, removes the original device notification, and replaces it with a new notification. The second notification shows either the unread email count or summary of the new emails, depending on your Email+ settings. To turn off detailed notifications, see **How do I turn on/off notification details on the lock screen?**.

Occasionally, the Email+ app is not able to sync new email in the background due to poor network connectivity or because the app is no longer running in the background. If this happens, you may continue to receive the first notification, which shows that you have new messages, but the second notification with the summary/unread email count will not display. To correct this, move to an area with better network connectivity and launch Email+.

Why am I not receiving Email+ Notifications?

There are a number of reasons why you may not be receiving notifications on your device:

- **Notification options are disabled**: Check the notifications options for Email+ in iOS device Settings to make sure that the options to "Allow Notifications", "Show in Notification Center", "Badge App Icon", and "Show on Lock Screen" are enabled.
- **Email+ app is force terminated**: If you force killed Email+ by flicking it off the top of the screen, the app will stop receiving the second notification. Please launch Email+ to start receiving notifications again.
- **Background App Refresh is disabled**: If you have disabled Background App Refresh for Email+, in your iOS device Settings, you will see the first email notification showing that you have new messages but will not see the second notification showing the new email details/unread email count. You can manage your background app refresh settings in your iOS device Settings by going to General > Background App Refresh.

- **Cellular Data option is disabled**: If the Cellular Data option is disabled and the device is not connected to WiFi, the device will not receive new email notification for Email+. Enable the Cellular Data option in your iOS device Settings for Cellular > Cellular Data and for Email+ > Cellular Data, to get notifications when the device is not connected to WiFi.
- **Device is in Low Power Mode**: If your device goes into Low Power mode when the battery is running low, Background App Refresh gets disabled and the second notification will stop working. Charge your device, disable low power mode, and launch Email+ to get notifications to work again.

iOS 9 introduced a new feature called Low Power Mode where the user can control whether a device can go into a battery conservation state to extend battery life. This is typically used when the device battery is running low and it is not possible to immediately recharge the device. In this state, iOS turns off background app refresh and also prevents apps from running in the background.

When this happens, Email+ will show the first notification ("You have new messages"), but will not be able to fetch updated unread email summaries in the background. So the second notification will not be displayed. To recover from this, charge your device fully and disable low power mode. You may also need to launch Email+ to get notifications to work again.

How do I turn on/off notification details on the lock screen?

You can control whether detailed notifications are displayed on the lock screen by using the Email+ notification settings in the iOS device settings. In addition, you can control whether the summary of new unread emails is displayed in the lock screen, by using the Show Mail Details option in the Notifications screen in the Settings section of the Email+ app.

If the Show Mail Details option is enabled, an individual notification is displayed for each unread email. If the option is disabled, a single notification that shows the aggregate count of unread emails is displayed.

Figure 3. Notifications options in the Email+ settings

AT&T
8:48 AM
82%
Settings
About
Sounds
Notifications
Mail
Contacts
Calendar
88
Mail
Calendar
Contacts
Tasks
Settings
Back
Notifications
ALERT TIMES
Work Hours Only
Weekend Alerts
VIP Allowed After Work Hours
The Alert Times options are disabled when real-time push notifications are enabled
MAIL ALERTS
Mail Alerts
All
Show Mail Details
You are subscribed to real-time push notifications. You will receive two notifications for every new message:
- The first notification from iOS will indicate that you have new message.
- The second notification from Email+ application will show the sender and subject of the message.
EVENT ALERTS

10. Rights Management System for iOS Overview

The Rights Management System (RMS) enables you to share encrypted mails to protect the content that is shared over email when using Microsoft Mail Exchange server.

When enabled the sender can control the distribution of the content shared over the mail. A rights managed email message is used to protect email content from inappropriate access, use, and distribution.

A rights policy template specifies whether a user can edit, forward, reply, reply all, print, extract (copy), export (remove protection), or programmatically access the content in the rights-managed email message. List of RMS permissions is displayed in **Email+ Settings/Troubleshooting/Available Permissions** and is selected in **New Mail Composer > lock button > Protect**. Due to ActiveSync protocol limitations, maximum of 20 RMS permissions are displayed in Email+.

Every protected mail has additional cell in mail viewer below subject cell, that contains information about license that is template name and description.

The admin can apply the following options to secure mail exchange as indicated by the **RightsManagementLicense** element included in the response. The RightsManagementLicense include:

- **ContentExpiryDate** - specifies the expiration date for the license (set to "9999-12-30T23:59:59.999Z" if the rights management license has no expiration date set).
- **ContentOwner** - specifies the email address of the content owner.
- **EditAllowed** - specifies if the content of the original email can be modified by the user when the user forwards, replies, or replies all to the email message.
- **ExportAllowed** - specifies if the IRM protection on the e-mail message can be removed by the user. The user can remove the IRM protection of the original message's content in the outgoing message when the user forwards, replies, or replies all to the original e-mail message;
- **ExtractAllowed** - specifies if the user can copy content out of the e-mail message (the content of the e-mail message can be cut, copied, or a screen capture can be taken of the content).
- **ForwardAllowed** - specifies if the user can forward the e-mail message.
- **ModifyRecipientsAllowed** – specifies if the user can modify the recipient list.
- **Owner** - value of **true** indicates that the authenticated user has owner rights on this message. This element is used for information presentation purposes only.
- **ProgrammaticAccessAllowed** - specifies if the contents of the e mail message can be accessed programmatically by third party applications.
- **ReplyAllAllowed** - specifies if the user can reply to all the recipients of the original e-mail message.
- **ReplyAllowed** - specifies if the user can reply to the e-mail message.
- **TemplateDescription** - This element is used for informational presentation purposes only.
- **TemplateID** - It contains a string that identifies the rights policy template.
- **TemplateName** - specifies the name of the rights policy template.

For more information on **To create a new Azure information protection template**, see [Microsoft documentation]

Setting permissions on an email

The permissions for email protection can be set on Email+ iOS app.

Setting permissions on Ivanti Email+ iOS app

Using the Email+ app to set email permissions.

Procedure

	1.	In the Email+ app, click on the compose mail icon.
	2.	Click on the **Lock** icon, select **Protect** option.
	3.	Select the permission you want to apply from the list of **Available Permissions** to the mail.
	4.	Click **Ok**.

Result: The selected permission is applied to the mail.

11. Searching mail in Ivanti Email+ app

Device users can search for mails in the Email+ app in particular folder or complete mailbox. You can preview mail from the search results list. Email+ iOS app supports token based search, you can select suggested token or any other that is automatically generated based on user emails. Type in the name of a sender or other keyword in the search field. The following default search tokens are suggested by the application:

- Unread Messages
- Flagged Messages
- Messages from VIP
- Messages with Attachments
- Messages with Calendar Invites

Multiple tokens can be applied to a search.

The following procedure describes how to initiate search for an email in a folder:

Procedure

1. In the Email+ app, select a mailbox.
2. Pull down the screen, search bar is displayed.
3. Type in the name of a sender or other keyword to in the search field.
4. By default, search is executed for “All Folders”. The search results view displays two tab and the search is executed by default in the "All Folders" view. The user can switch to "Current Folder" if required.

12. Introduction to Ivanti Email+ Notification Services

When a mail is received on an iOS mobile device, a notification appears if real-time notification is enabled. For each mail there are two notifications received:

- **Apple APNS notification**: Shows up immediately on the lock screen (depending on the Email+ Notification settings in your device Settings). This notification has the text: “You have new messages”.
- **Email+ notification**: The Email+app fetches the email summary for the new unread email, removes the original device notification, and replaces it with a new notification. The second notification shows either the unread email count or summary of the new emails, depending on your Email+ settings.

VoIP notifications for Email+ are not supported starting Email+ 3.13.0 and later, due to the changes in Apple Policy with regards to notifications. This impacts the **VIP notifications outside of work hours**, other notifications such as **Work hours notifications** and the **Weekend notifications** work as before.

Email+ Notification Services (ENS) is the recommended solution to fix this issue.

There are two different notification deliverable mechanism for real-time notification:

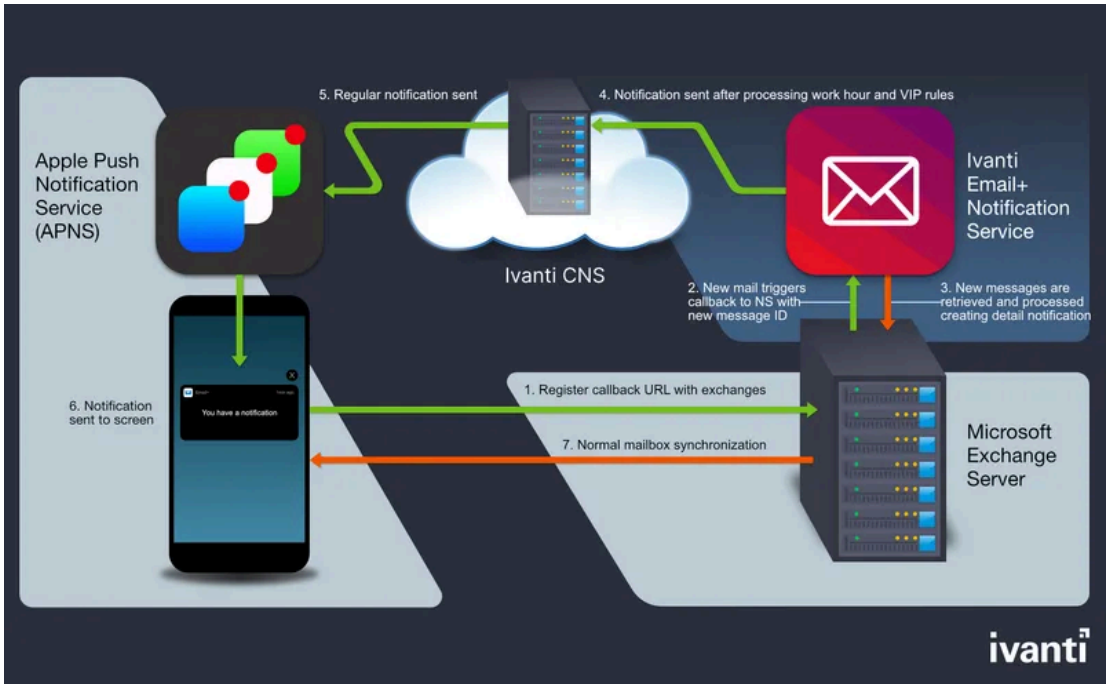
1. Cloud based service which only supports work hour notification, see **About real-time push notifications for Email+ for iOS** section.
2. ENS based service which supports VIP email only notifications. The following sections describes about ENS and how to configure it on your device.

About Ivanti Email+ Notification Services

Email+ Notification Service leverages a standalone server that is based on existing Sentry and hosted inside the organizations firewall. This server is configurable on UEM servers and leverages a service account on Microsoft Exchange to retrieve email metadata information (such as Sender, Subject, and Body Snippet) required to support VIP mails notifications outside of office hours and direct detail notifications.

The following diagram describes how the ENS solution works:

Figure 1. Email+ Notification Services Architecture



1. **Email+ App**: The Email+ app registers callback URL with Microsoft Exchange server to send out work hours, list of VIP contacts, key-value pairs settings, and VIP notifications outside work hours.
2. **Microsoft Exchange**: When a new mail is received in the folder that you are subscribed to, Microsoft Exchange sends a notification to Email+ Notification Proxy (ENP) which consists of Message Id and Folder Id for the new received mail.
3. **Email+ Notification Proxy**: ENP logs in to the exchange server using a service account and pulls the following information about the mail: Senders email address
    1. Subject line
    2. Snippet of the mail
    3. ENP applies and verifies the rules. Once the verifications is complete, it creates a payload and sends it to real-time push notification server on the Ivanti Neurons for MDM.
4. **Real-time push notifications**: The MobileIron CNS relays the information to Apple Push Notification service (APNs).
5. **Apple Push Notification service (APNs)**: Notifies the iOS device.
6. **Notification workflow on Email+**: This feature requires users to be subscribed to CNS for Real Time Notifications. iOS displays a notification to the user indicating that there are new messages.
7. **Mailbox active sync**: A notification is triggered to Email+, to open the correct mailbox a sync up is performed to co-relate the notification and the mail.

Enable Email+ settings on your device to receive notifications. In Email+ app, go to **Settings > Notifications** and enable **Work Hours Only** option first to enable **Weekend Alerts** or **VIP Allowed After Work Hours** option. The VIP related options are not be enabled if ENS is not configured, as there is no service account to check if the sender is a VIP contact (ensure that the contacts are marked as VIP).

When notification settings is changed, a note similar to the following is displayed:

Changes to alert times will come into effect after 'X' hours. The default interval is 1440 minutes.

> If you are using CNS only for real-time notifications. all the settings related to VIP are not visible on the Email+ app and cannot process notifications outside work hours other than delivering notifications over weekend.

The following section describes how to configure ENS.

Limitation: The ENS solution is not supported on Microsoft Exchange Office 365.

Before you begin

- Supported on Ivanti EPMM 10.7.0.0 or later, Ivanti Neurons for MDM R70 and later, and Sentry 9.8.5.
- Ensure that you have configured a service account on Microsoft Exchange Server (Service account on exchange impersonates other mailboxes when accessing exchange over various supported protocols. For the purpose of Exchange Notification Proxy (ENP), Microsoft's Exchange Web Services (EWS) protocol is used to access mailbox messages.)
- Ensure that you have the JWT token of CNS production server. For more information, see **About real-time push notifications for Email+ for iOS** section.

> The term JWT token is also referred as **Authorization Token**, **Token**, and **notification_server_authorization** in Ivanti products.

- Standalone Sentry must be configured ActiveSync with a publicly trusted certificate.
- Ensure that the Exchange servers are configured with the service account. The servers must have identity certificate to authenticate the service account
- If Exchange server version support is earlier than TLS v1.2, then the supported protocols should be configured in Incoming protocols on MICS.

The following table describes the ENS port rules for firewall.

Table 1. ENS port network rules

RequireDirectionment	Destination	Port	Direction
Standalone Sentry	CNS.mobileiron.com	TCP443	Outbound Initialized, Bi-direction Connection
Exchange Server	Standalone Sentry	TCP443	Outbound
Standalone Sentry	Exchange Server	TCP443	Inbound
Ivanti EPMM Management IP	Standalone Sentry	TCP443, 9090	Bi-direction Connection
IT Admin PC IP	Standalone Sentry	TCP8443, 22	Inbound Initialized, Bi-direction Connection
Standalone Sentry	NTP Server	NTP	Outbound
Standalone Sentry	DNS Server	DNS	Outbound
Standalone Sentry	SMTP Server	SMTP	Outbound

Configuring service account

Service account on Microsoft Exchange impersonates other mailboxes when accessing exchange over various supported protocols. Following are the main steps for configuring service account.

- Setting up service accounts on Exchange server
- Configuring a service account on Exchange server

Setting up service accounts on Exchange server

For the purpose of Exchange Notification Proxy (ENP), Microsoft's Exchange Web Services (EWS) protocol is used to access mailbox messages.

For example service account is assigned to the following role:

ApplicationImpersonation

The EWS sends requests with the credentials of a single service account which includes an .XML key.

<soap:Header>

<t:RequestServerVersion Version="Exchange2013" />

<!-- The following causes the request to run as alfred@contoso.com —>

<t:ExchangeImpersonation>

<t:ConnectingSID>

<t:SmtpAddress>alfred@contoso.com</t:SmtpAddress>

</t:ConnectingSID>

</t:ExchangeImpersonation>

</soap:Header>

This allows a single account to access the mailbox of other accounts.

Configuring a service account on Microsoft Exchange Server

To configure service account on EWS follow these steps:

1. In the Microsoft Exchange Management console, open a browser and type in URL. For example:*https://<hostname>/ecp*
2. Log in as an Admin, go to **Mail > Options > Manage My Organization > Roles & Auditing> Mailboxes** and create a new Role group.
3. Add the **applicationImpersonation** role to the group.
4. Add members to the group.
5. Click **Save** to finish.

For more information on configuring service account on Microsoft Exchange server, see **Microsoft documentation**

Setting up Standalone Sentry as an Ivanti Email+ Notification Service

You can set up a dedicated Standalone Sentry as an Email+ Notification Service. This capability allows you to configure multiple Exchange servers to provide notifications for VIP accounts in Email+.

This feature requires UEM servers, Cloud Notification Service (CNS), Standalone Sentry, and Email+. Applicable to iOS only, the Email+ Notification Service cannot be combined with ActiveSync or AppTunnel. Email+ Notification Service requires Sentry 9.8.5 and

Email+ 3.13.0 through the latest supported versions. (Content Notification System is automatically upgraded). For more information, see: "*Standalone Sentry Email Notifications*" section in the Ivanti Sentry Guide for Ivanti EPMM

Configuring Ivanti Email+ using KVPs on Ivanti EPMM for Notification Services

After Standalone Sentry is set up, you must configure Email+ on Ivanti EPMM.

Procedure

1. In the Ivanti EPMM Admin Portal, go to **Policy & Configs > Configurations**.
2. Click **Add New > AppConnect > App Configuration** to create a new AppConnect configuration.
3. In the Name field, enter brief text that identifies this AppConnect app configuration. For example: Email+ for iOS.
4. In the Description field, enter additional text that clarifies the purpose of this AppConnect app configuration.
5. In the Application field, enter the bundle ID for the app:
6. In the **App-specific Configurations** section enter the required key-value pairs. For more information on how to configure the key-value pair on Cloud Notification Service, see Real-time push notifications section.

- **com.mobileiron.ios.emailplus**

7. Click **Save**.
8. Go to **Policies & Configs > Policies**, select an AppConnect policy and click **Edit**. Guidelines to edit AppConnect Global Policy:

Fields	Option
Name	Default AppConnect Global Policy
AppConnect	Select **Enabled** option
Security Policies > Apps without an AppConnect container policy	Check the **Authorize** option

9. Click **Save**.
10. Go to **Apps > App Catalog > Add + >In-house**.
11. Click **Browse** to **Upload Email+ Inhouse App**.

Registering your iOS device using Ivanti EPMM

You should register your iOS device with an LDAP user or local user:

1. In the Ivanti EPMM Admin Portal, go to **Device & Users > Users**.
2. Click on **Add** and select **Add Local User** or **LDAP User**.
3. Fill in the details in the **Add New User** window.
4. Click **Save**.
5. Register device with the Local or LDAP user.
   Result: Email+ is pushed to Device as a part of MDM configuration.

Configuring Ivanti Email+ using KVPs on Ivanti Neurons for MDMfor notification services

Set up a Standalone Sentry before configuring Email+ on Ivanti Neurons for MDM, see the Standalone Sentry Email+ Notification Services section in the Ivanti Sentry Guide for Ivanti Neurons for MDM.

Procedure

1. In the Ivanti Neurons for MDM Admin Portal, go to **Apps > App Catalog > Email+**
2. Go to **App Configuration > Email+ Configuration**, click **+** to create a new Email+ configuration.
3. In the **Configuration Setup** section enter the following:
4. In the **Name** field, enter the name of the configuration.
5. In the **Description** field, enter additional text that clarifies the purpose of the configuration.
6. In the **Email+ Settings** section, enter the following:
    1. Email Address
    2. Exchange Host
    3. Exchange Username
7. In the **AppConnect Custom Configuration** section enter the required key-value pairs, to configure ENS.
8. Choose a distribution option for the configuration and click **Done**. The configuration is distributed to the subset of the devices to which the app is distributed.

> The **sentry_server_host** key-value pair should point to ENS Sentry hostname.

Registering your iOS device using Ivanti Neurons for MDM

To register your iOS device with Email+, see the Device Registration (iOS, macOS, and Android) section in the Ivanti Neurons for MDM Administrator Guide.

Procedure

1. In the Ivanti Neurons for MDM Admin Portal, go **Admin >User**.
2. Click on **+Add** to add a user, select the user type.
3. Click **Done**. New user is added.
4. In the Go Client, log in with the user details. After registering on Go Client. The device is listed under the devices tab in Ivanti Neurons for MDM.

13. ActiveSync16 for iOS devices

Ivanti Email+ supports "**ActiveSync 16**" protocol for iOS devices. The "**ActiveSync 16**" protocol allows mobile devices to synchronize email and other data with Microsoft Exchange mailbox. The ActiveSync 16 protocol supports latest Exchange platform starting from Exchange 2016 and Exchange Online.

> Starting from Email+ 4.15.0 version, ActiveSync 16 is enabled by default on all the iOS devices that are supported by Email+.

Previously, Microsoft announced **Draft** Folder synchronization for the first time with ActiveSync protocol version 16. The capability to sync draft messages between devices is now available on the Email+ app.

- Email+ 4.1.0 for iOS provides an early preview of the following capabilities that are now supported through ActiveSynce16 protocol:
  - **ActiveSync 16 support**: Sets up the synchronization engine to use ActiveSync 16 protocol.
  - **Display BCC contacts in Sent items**: Allows the sender to view BCC contacts in sent emails.
  - **Calendar attachments**: Calendar attachments are synced through ActiveSync 16 protocol instead of EWS. However, the **email_ews_host** key-value pair is still required for notifications.
  - **Drafts folder synchronization**: Enables access and editing of emails in the **Drafts** folder from all the devices synchronized with Microsoft Exchange server.

ActiveSync 16 is now enabled by default, to disable **eas_16** value add it to the "**disabled_features**" key-value pair. The following features are available with **eas_16**:

- Displays BCC contacts in **Sent** items
- Syncs Calendar attachments through ActiveSync 16 protocol instead of EWS. With **eas_16** enabled, "**calendar_attachments**":"true" and "**email_ews_host**":"<host>" KVPs are no longer required to see event attachments
- Syncs Drafts with the server If the server doesn't support ActiveSync 16.0 or newer, mentioned features won't be enabled and Email+ app will work as per the current settings.

14. Delegated calendar

Ivanti Email+ supports delegated access for Calendar. The delegated calendar option enables the calendar owner to assign their calendar to a delegated user within the organization's exchange Global Address List (GAL). To enable the **Add Delegated Mailbox** option in Email+, configure the **calendar_delegation** value in **enabled_features** key value pair.

Microsoft Exchange server provides Reviewer, Editor, and Author level permissions, in the Email+ app delegated user can create, view edit and remove calendar events according to permissions given by the owner. The delegated user can view event related updates on the Email+ app with this permission.

The following table displays the different delegation permissions and the actions they can perform:

Permission level	Action allowed
Reviewer	With the Reviewer level permissions you can perform the following actions: • Appointments (event without attendees) - no options • Delegated calendar owner's created invite - no options • Received invite – Reply / Reply All / Forward on behalf of
Author	With the Author level permissions, you can perform the following actions: • Appointments (event without attendees) - no options • Delegated calendar owner's created invite - no options • Received invite – Reply / Reply All / Forward on behalf of • Author created appointments – no Reply / Reply All / Forward options. Edit button is displayed • Author created invites – no Reply /Reply All / Forward options. Edit button is displayed
Editor	With the Editor level permissions, you can perform the following actions: • Appointments (event without attendees) - no Reply / Reply All / Forward options. Edit button is displayed • Delegated calendar owner's created invite – no Reply / Reply All / Forward options. Edit button is displayed • Received invite – Reply / Reply All / Forward on behalf of • Editor created appointments - no Reply / Reply All / Forward options. Edit button is displayed • Editor created invite - no Reply /Reply All / Forward options. Edit button is displayed

To configure delegation, configure the following key-value pairs:

- Add **calendar_delegation** value to **enabled_features** key-value pair to add the **Add Delegated Mailbox** option in the Email+ app.
- Add **email_ews_host** key-value pair with EWS host value as FQDN to provide access to EWS server.
- Add **ews_min_allowed_auth_mode** key value pair to enable basic, certificate based authentication, and modern_auth authentication methods to the exchange server through EWS protocol. If Email+ is configured with **eas_min_allowed_mode** KVP as Modern Auth, add **ews_min_allowed_auth_mode** KVP with **modern_auth** value to enable modern authentication method to the EWS server.
- If EWS server is not accessible publicly (located in private network), then VPN should be configured.

When a calendar is delegated, a system generated mail is sent to the delegated user with details of the calendar owner. The calendar owner has the ability to assign or deny access to the delegated calendar.

Adding delegated calendar

In the Email+ app, go to **Settings > Accounts > Add** option and enter the calendar owner's email address. Select the calendar owner to add the delegated calendar. You can assign a color to the delegated calendar to differentiate between assigned calendars.

> The calendar owner delegates the calendar from the Microsoft Outlook on the web (OWA). For more information on delegation, see **Microsoft documentation**.

The user can manually remove the delegated calendar from the Email+ app. If the access to the deleted calendar is available, then user can again add that delegated calendar.

When the calendar owner removes access to the delegated calendar, Email+ receives "**Delegated Calendar access has been denied**" notification when the Email+ app is launched.

Disabling delegated calendar feature

The admin can disable the **Add Delegated Mailbox** option from the Email+ app by removing **calendar_delegation** value from the **enabled_features** key value pair.

Ivanti Email+ configurations supported for Delegated Calendar

The following table lists the supported Email+ and EWS configurations.

> If EWS server is not accessible publicly (located in private network), then VPN should be configured.

Email+ Configuations	Additional Configurations for EWS	Supported
Ivanti EPMM and Ivanti Neurons for MDM with sentry, Modern auth with or without **email_password** KVP, Microsoft Office 365	Add **email_ews_host** KVP with EWS server value Add **ews_min_allowed_auth_mode = modern_auth** KVP	Yes
Ivanti EPMM or Ivanti Neurons for MDM, without sentry, Modern auth with or without **email_password** KVP, Microsoft Office 365	Add **ews_min_allowed_auth_mode = modern_auth** KVP	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with sentry, Modern auth with or without **email_password** KVP, Microsoft Office 365	**Exchange host for EWS** should have value of the EWS server Add **ews_min_allowed_auth_mode = modern_auth** KVP	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with sentry + Local certificate, Basic auth with or without **email_password** KVP, Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM, with sentry + group certificate, Basic auth with or without **email_password** KVP, Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM, with Sentry + MS scep certificate, Basic auth with or without **email_password** KVP, Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with Sentry, Kerberos with **prompt_email_password=true** and enter password on Email+ login screen, Microsoft Exchange versions 2016 and 2019	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with Sentry, Kerberos with **email_password** KVP with hard coded value (which is not probably a use case), Microsoft Exchange versions 2016 and 2019	Add **email_ews_host** with EWS server value.	Yes
Ivanti EPMM or Ivanti Neurons for MDM, without sentry, Basic auth (with or without 'email_password' KVP), Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value.	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with or without sentry, Certificate based auth, Microsoft Exchange versions 2016 and 2019	Add **email_ews_host** with EWS server value Add **ews_min_allowed_auth_mode = cert_base** KVP .	Yes

Email+ Configuations	Additional Configurations for EWS	Supported
	For **cert_base** KVP, Microsoft Office 365 is not supported.	
Ivanti EPMM or Ivanti Neurons for MDM, with sentry, Kerberos (without 'email_password' KVP), Microsoft Exchange versions 2016 and 2019		No

Email+ does not show alerts and notifications for delegated events.

15. Delegated mailbox

Ivanti Email+ supports delegated access for Mailbox. Currently, we support delegation of up to four mailboxes. In case of multiple delegated mailboxes, each mailbox is independent and has its own Delegated Mailbox account settings.

When configuring the mailbox delegation, the owner delegates mailbox to another user (Delegate) to manage it as per delegation level set on Outlook. The delegated user can add up to four delegated mailboxes from multiple owners.

In the Email+ app, under the **Calendar tab > All Calendars screen**, the shared accounts now have a **Display Name** along with the email address.

To configure Delegated Mailbox use the following key-value pairs:

- Add **delegated_shared_mailbox** value to **enabled_features** key-value pair to enable the **Add** button in the **Settings > Mail > Accounts** section of the Email+ app.

When you select **Add** option and enter details **Email ID** field, Email+ searches email address in GAL and provides list of contacts to add mailbox. If the search result is successful, the **'Add**' option on the **Add Delegated Mailbox** screen is enabled. Click on the **Add** option to add delegated mailbox.

The delegated mailbox permissions are similar to Microsoft Exchange server such as **Reviewer**, **Author**, and **Editor** level permissions. Starting Ivanti Email+ 4.9.0 all permissions levels are supported in Email+.

Delegated user can now save draft mail in the Delegated Mailbox. The Delegated Drafts folder is not synced with the server and only available locally. Delegate can store draft emails in "**Drafts**" folder. These folders are local and are not synced with the server.

If Delegated Calendar is added without Delegated Mail, then the draft emails (such as reply/forward for delegated events) are saved in "**Delegated Drafts**" folder for Primary account. This folder is also local and is not synced with the servers.

- Search is not valid for "**Drafts**" in Delegated Mailbox and in the "**Delegated Drafts**".
- Mails cannot be moved from or to Delegated Mailbox Drafts and "**Delegated Drafts**".

The mailbox owner can change permissions from any role (Reviewer/Author/Editor) to any role (Reviewer/Author/Editor), also for existing drafts.

- When the **Owner** revokes access to Delegated Mailbox completely, or the user removes the Delegated Mailbox manually from the Email+ settings, related Drafts folder is removed with all its content.
- When the **Owner** revokes access to Delegated Calendar, all related drafts from the "Delegated Drafts", Delegated Mailbox "Drafts", and Outbox (if any emails are stuck there) folders are removed.
- When the **Admin** removes the **delegated_shared_mailbox** key-value pair from the Email+app config, then all Delegated Mailbox drafts are deleted from the app along with the other information from the Delegated Mailbox.

The mailbox owner can update the following permission combinations:

Permission level	Combinations
Reviewer	• Reviewer > Author • Reviewer > Editor • Reviewer > None
Author	• Author > Reviewer • Author > Editor • Author > None
Editor	• Editor > Author • Editor > Reviewer • Editor > None

The following table displays the different delegation permissions and the actions they can perform:

Action	Reviewer	Author	Editor
Reading emails	Yes	Yes	Yes
Downloading and viewing email attachment	Yes	Yes	Yes
Searching emails	Yes	Yes	Yes
Marking emails as Read/Unread	Yes	Yes	Yes
Flagging emails	No	No	Yes
Moving an email to a different folder within the account	No	No	Yes
Deleting emails	No	No	Yes
Sending emails with attachments	No	Yes	Yes
Options to Reply/Reply All/Forward emails	No	Yes	Yes
Saving draft emails locally in the app. *Reviewer cannot compose emails and create drafts. The only possible way to create draft is an edge case when doing Reply/Reply All/Forward a delegated calendar event.	Yes	Yes	Yes
Responding to meeting invites from emails (no delegated calendar added)	No	No	No
Responding to meeting invites from emails (DC added). * Becomes available if the Delegated Calendar of the same owner's account is added with Author or Editor permissions.	Yes	Yes	Yes
Reading signed and/or encrypted emails	Yes	Yes	Yes
Sending signed and/or encrypted emails	No	Yes	Yes
Reading emails with Classification	Yes	Yes	Yes
Sending emails with Classification	No	Yes	Yes

The delegated user can manually delete the delegated mailbox added from the Email+ app.

> Drafts are stored locally and are never synced with the server. Even if **eas_16** is enabled for the Primary account.

> The root folders created by the user are not supported and not accessible. This is a limitation of the EWS protocol.

> Not possible to add Shared mailbox if account root folder was shared

For interval-based notification, the folder sync is triggered for the next available EWS delegated mailbox to ensure all mailboxes have equal priority. Each mailbox has its own queue of folders for synchronization, allowing for more balanced and efficient syncing. It is not recommended to enable automatic sync for many folders and this will increase the full sync cycle.

Adding delegated mailbox to the Email+ app

To add the delegated mailbox to the Email+ app on your iOS device, perform the following steps:

Before you begin

- Configure the enabled_features key-value pair and add the **delegated_shared_mailbox** value.
- Set both EWS and Exchange host
- Set EWS Authorization mode,
  - For Modern Auth, set value to **modern_auth**
  - For Certificate based Auth, set value to **cert_base**

Procedure

1. In the Email+ app, go to **Settings > Accounts > Add**.
2. On the **Add delegated Mailbox** screen, type the **Email ID** of the mailbox owner. Delegated mailbox is added in Mail section - Mailboxes. The delegate gets access to Inbox, sub folders, and smart folders.

Support for SMIME (Early preview)

The signing and encryption functionality is extended to support Delegated Mailboxes, the functionality works similar to that of Primary Account. Delegated user can:

- Read/Send signed/encrypted emails depending upon the user permission
- Read emails with Classification

The keychain is common for all mailboxes and not only for the mailbox you are working on. The app searches certificates in the keychain and GAL (same as the primary account).

The process of adding certificates to the Keychain is:

- Through **email_signing_certificate** and **email_encryption_certificate** KVPs.
- From the email attachment

Signing

A Delegate can view, search, reply, and forward signed emails in the delegator's mailbox (depending on permissions).

In the received email, the sender's signing certificate can be verified if their public certificate is available in GAL, or if it is available in the delegate's keystore in Email+. Otherwise, the certificate will be marked red as not trusted.

A Delegate can sign emails with the actual sender's certificate (Primary account) and send signed emails from the Primary account on behalf of the delegator.

If the certificate of the signed email cannot be validated (there is no user certificate in the keychain and in GAL), the red check mark icon is be displayed according to the existing logic.

Encryption

A Delegate can encrypt emails with the actual sender's certificate (Primary account) and send encrypted emails from the Primary account on behalf of the Delegator.

A Delegate can decrypt emails if the Delegator provides the Delegate with their private certificates, and these certificates are added to the app Keychain.

Suppressing Name Check on certificate mismatch

When the feature flag for suppressing name checks on certificate mismatch is enabled by admin, the feature is available for delegated mailboxes. Existing certificate associations are automatically accessible for delegated mailboxes.

The user can encrypt/decrypt emails using existing certificate associations and create new associations.

Classification

When classification is configured by Admin, it is available for delegated mailboxes. The user is able to parse emails in the delegated mailboxes having classification, reply/forward, and compose new emails.

Ivanti Email+ configurations supported for Delegated and Shared Mailbox

The following table lists the supported Email+ and EWS configurations.

> If EWS server is not accessible publicly (located in private network), then VPN should be configured. Ensure that the user has access to the EWS host specified in the KVP from the device. If EWS host is not reachable, contact your admin to setup connection using the VPN to the internal network where the EWS host is accessible.

> Update the host name in the **email_ews_host** key-value pair.

Email+ Configuations	Additional Configurations for EWS	Supported
Ivanti EPMM and Ivanti Neurons for MDM with sentry, Modern auth with or without **email_password** KVP, Microsoft Office 365	Add **email_ews_host** KVP with EWS server value Add **ews_min_allowed_auth_mode = modern_auth** KVP	Yes
Ivanti EPMM or Ivanti Neurons for MDM, without sentry, Modern auth with or without **email_password** KVP, Microsoft Office 365	Add **ews_min_allowed_auth_mode = modern_auth** KVP	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with sentry, Modern auth with or without **email_password** KVP, Microsoft Office 365	**Exchange host for EWS** should have value of the EWS server **EWS Authentication Mode** should have **Modern Authentication** value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with sentry + Local certificate, Basic auth with or without **email_password** KVP, Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM, with sentry + group certificate, Basic auth with or without **email_password** KVP, Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM, with Sentry + MS scep certificate, Basic auth with or without **email_password** KVP, Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with Sentry, Kerberos with **prompt_email_password=true** and enter password on Email+ login screen, Microsoft Exchange versions 2016 and 2019	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with Sentry, Kerberos with **email_password** KVP with hard coded value (which is not probably a use case), Microsoft Exchange versions 2016 and 2019	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, without sentry, Basic auth (with or without 'email_password' KVP), Microsoft Exchange versions 2016 and 2019, Microsoft Office 365	Add **email_ews_host** with EWS server value	Yes
Ivanti EPMM or Ivanti Neurons for MDM, with or without sentry, Certificate based auth, Microsoft Exchange versions 2016 and 2019	Add **email_ews_host** with EWS server value Add **ews_min_allowed_auth_mode = cert_base** KVP	Yes

Email+ Configuations	Additional Configurations for EWS	Supported
Ivanti EPMM or Ivanti Neurons for MDM, with sentry, Kerberos (without 'email_password' KVP), Microsoft Exchange versions 2016 and 2019		No

16. Shared mailbox

Ivanti Email+ supports shared access for Mailbox. In case of multiple mailboxes, each mailbox is independent and has its own Shared Mailbox account settings. A Shared Mailbox is a type of user mailbox that does not have a username or password. It provides access to multiple users to use a mailbox as per the permissions set by the admin on the mail server (Office 365 or Exchange). To access a Shared Mailbox, the admin must add the user as a member of the mailbox. Email+ supports Shared Mailbox permissions similar to Microsoft Exchange server:

- **Full Access**: The full access permission allows the user to open the Shared Mailbox and act as the owner of that mailbox. The user can access mails, read, view, delete, move mails, download and view attachments, and search mails on the local device and server. However, the user cannot send mail
- **Send as**: The send as permission allows the user to impersonate the Shared Mailbox when sending mails. The user can Reply or Reply All, and Forward messages from the Shared Mailbox.
- **Send on behalf**: The send on behalf permission allows the user to send mail on the behalf of the Shared Mailbox. The user can Reply or Reply All, and Forward messages from the Shared Mailbox, but the recipients would receive the mail from <User on behalf of the Shared Mailbox name>.

The user can add signature to mails for both delegated and shared mailbox similar to the primary mail box, this signature is configured and stored locally in the app, and cannot be set in the Email+ config like it is for the Primary account. Separate section for **Shared Mailbox** and **Delegated Mailbox** accounts is added under the **Email+ Settings → Mail → Accounts**. The settings for these two sections appear only if there is at least one account is available for each mailbox.

In the Email+ app, under the **Calendar tab > All Calendars screen**, the shared accounts now have a **Display Name** along with the email address.

The Email+ app displays a notification for both interval-based and real-time notifications. New Shared mailboxes option is added to Settings → Notifications → Mail Alerts.

For interval-based notification, the folder sync is triggered for the next available EWS shared mailbox to ensure all mailboxes have equal priority. Each mailbox has its own queue of folders for synchronization, allowing for more balanced and efficient syncing. It is not recommended to enable automatic sync for many folders and this will increase the full sync cycle.

Support for SMIME and Classification

The Email+ app now supports SMIME and classification for the shared mailbox. The Signing and Encryption functionality is similar to Primary or Delegated mailbox. For more information, see the **Support for SMIME (Early preview)** section under **Delegated mailbox**

Configuring shared mailbox

To configure Shared Mailbox on, configure the following key-value pairs:

- Add **delegated_shared_mailbox** value to **enabled_features**key-values.

When you select the **Add a Mailbox** option, Email+ searches email address in GAL and provides list of contacts to add mailbox. If the search result is successful, if the user has a permission to access the account being searched, then the "**Add**" button becomes available and the user can add the **Shared Mailbox**.

Adding shared mailbox to the Email+ app

To add the shared mailbox to the Email+ app on your iOS device, perform the following steps:

Before you begin

- Configure the enabled_features key-value pair and add the **delegated_shared_mailbox** value.
- Set both EWS and Exchange host
- Set EWS Authorization mode,
  - For Modern Auth, set value to **modern_auth**
  - For Certificate based Auth, set value to **cert_base**

Procedure

1. To add new mailbox, go to **Email+ Settings > Accounts > Add** , On this screen, type the **Email ID** of the delegated or shared mailbox.
2. If the email search result is successful and the user has a permission to access the account being searched, then the "**Add**" button becomes available and the user can add the **Shared Mailbox**.

For more information on supported configuration on shared mailbox, see **Ivanti Email+ configurations supported for Delegated and Shared Mailbox** section **Delegated mailbox**.

17. Personal calendar

Search enabled for personal calendar, the Ivanti Email+ app now supports search in Calendar for personal events. The following features are available:

- Provision to display all the calendars available in iOS native calendar in the Email+ Calendars screen with a read-only view.
- All kinds of events are supported - single, recurrent, all-day, several days, invites.
- The Calendar of a particular email account will be indicated on each view in different colors.
- When a new event is created or updated or deleted, including events responses from the personal Calendar, changes are applied to Email+ Calendar as well.

18. Pre-fill an email in the Compose view in Email+

Composing emails was extended with the new "**email+app**" scheme with the same variables as for mailto scheme:

email+app://compose?to=some@example.com&subject=testSubject

When you tap on any URL from any app, the Email+ app is activated and Composer window opens with pre-filled fields.

Link Syntax:

Domain: email+app

Host: compose

Variables:

Variable	Description
&to=	Use this variable to set the recipient, or recipients, separated by a comma.
&cc=	Use this variable to set the CC recipient(s).
&bcc=	Use this variable to set the BCC recipient(s).
&subject=	Use this variable to set the email subject, URL encode for longer sentences, so replace spaces with %20, and so on.
&body=	Use this variable to set the body of the message, you can add entire sentences here, including line breaks. Line breaks should be converted to %0A.

19. Configuring Azure Conditional Access (Optional)

The Azure Conditional Access validates user, device, location, and risk level before granting or denying access. The Email+ app is migrated from Azure Active Directory Authentication Library (ADAL) to Microsoft Authentication Library (MSAL) to support Azure Conditional Access.

> Optionally, configure Azure Conditional Access to restrict non compliant devices and unmanaged Microsoft 365 accounts.

Creating a new Conditional Access policy

The following procedure describes how to create a new policy using Azure admin portal:

Procedure

1. In the Azure admin portal, login as admin.
2. Go to **Security > Protect > Conditional Access**.
3. In **Conditional Access** settings, click **Create new policy** to create a new policy.
4. In **New** Conditional Access Policy page, update the following fields to create a new policy:

Fields	Option
Name	Enter name of the policy.
Assignments **Target resources**	Controls the access based on who the policy will be applied to, such as users and groups, workload identities, directory roles or external guests, and so on. Under **Users** select: • **Include** • **Exclude** Select **Email+Auth**
Conditions	Select device platforms : • **Android** • **iOS**
Access controls	**Grant**: Controls access enforcement to block or grant access. Under **Grant** section, select the following as required: • **Grant access**: Select require device to be marked as compliant.
Enable Policy	Toggle and select from the following options: **Report-only**: Policies in Report-only more requiring compliant devices may prompt users on macOS, iOS, Android, and Linux to select a device certificate • **On** • **Off**

5. Click **Create** to save the policy.

Editing a Conditional Access policy

The following procedure describes how to edit the existing policy to configure Azure Conditional Access using the Email+Auth:

Procedure

1. In the **Azure admin portal**, login as admin.
2. Go to **Security > Protect > Conditional Access > Policies.**
3. Select and edit the policy, and update the following:
    1. Add **Email+ Auth** application as a target resource.
    2. Go to **Access Controls > Grant** and select the **Require device to be marked as compliant** option.

The Email+ application is now set up for Azure Conditional Access to support complaint devices.

20. Microsoft Teams Integration in Email+ Calendar

Ivanti Email+ now supports Microsoft Teams integration in Email+ Calendar for iOS devices. This feature enables you to add Microsoft Teams meeting links directly into your calendar events with a toggle, removing the need to switch between different applications such as Outlook or the Teams application. Currently, support is available only for primary accounts.

The following features are available:

- The admin can enable or disable this feature.
- You can choose whether all new events should include a Teams meeting link by default, or add them only when needed. For more information see, **Setting your preferences**.
- The meeting details are added to the event notes.

> The Microsoft Teams Integration in Email+ Calendar feature works only for Microsoft 365 users using Modern Authentication. Basic and Certificate based authentication modes are not supported. It is not available for Exchange Server users.

Configuring Microsoft Teams in Email+ Calendar

To configure Microsoft Teams, use the following key-value pairs:

- Add **microsoft_teams_meetings** value to **enabled_features** key-value pair to enable the toggle button in the **Calendar> New Event> Teams Meeting** section of the Email+ app.

As Graph API calls are used for this teams integration, the following KVPs apply:

- **email_graph_host**: Value is graph.microsoft.com.
- **msal_graph_resource_scope**: Value is https://graph.microsoft.com/default.
- **msg_min_allowed_auth_mode**: Value is modern_auth.

Setting your preferences

To set your preferences, perform the following steps:

Procedure

- Go to **App Settings > Calendar Settings**.
- In the **Online Meeting** section, use **Make Every Meeting Online** toggle to set your default:
  - **ON**: Newly created events will include a Microsoft Teams meeting link by default.
  - **OFF**: A Microsoft Teams meeting link can be added to individual events as needed.

> **Make Every Meeting Online** section is available when the **microsoft_teams_meetings** KVP is enabled. By default, it is switched on.

Scheduling a meeting and add Invitees

To schedule a meeting and add invitees in the Email+ app, perform the following steps:

Procedure

1. In the Email+ app, click **Calendar**.
2. From the Calendar, select **New event**.
3. Add a title for your meeting or event.
4. Select a suggested time slot, or manually enter a date, start and end times in the **Start** and **End** sections.
5. Click **Invitees** .
   In **Add Invitees** enter names of individuals to invite to the meeting or event. Email+ will provide auto populated suggestion based on your recent interactions. Choose a name from the list to add as an invitee.
6. Select the toggle button to add a Teams meeting link in the **Teams Meeting** section.

> By default, the Teams meeting toggle is switched off. It is enabled only when the **Make Every Meeting Online** section is switched on in calendar settings.
> When an event is created without adding invitees, the Microsoft Teams toggle is not automatically enabled, and no Teams invite is generated for the event.
> When invitees are added, the Microsoft Teams toggle is automatically enabled, and a Teams invite is generated for the event.

1. Click **Done**.
2. Click on the created event or meeting in the **Calendar**.
   You can view the details of the event or meeting and the Teams meeting link.

Editing a teams meeting

To edit a meeting perform the following steps:

Procedure

- In the Email+ app, click **Calendar**.
- Open the existing meeting and click **Edit**.
- You can add or remove any meeting details.
- Click **Done**.
  The changes will reflect in the Microsoft teams application.

> If the event already has Teams details, the toggle may be disabled to prevent accidental removal.

Joining a teams meeting

To join a meeting perform the following steps:

Procedure

- In the Email+ app, open the event in your calendar.
- Click **Teams meeting** link in the Notes section, or use the **Join** button

The meeting will open in your browser or the Microsoft Teams app (if installed).

> The **Join** button is displayed when either of the following KVPs is enabled: **allow_universal_links** or **allow_safari_browser**.

Limitations

- Delegated Accounts: Teams integration is not available for delegated mailboxes.
- Other Meeting Providers: Only Microsoft Teams is supported at this time.
- Exchange Servers: Not supported due to technical limitations.
- Localization: Meeting details in event notes are handled by server settings.

21. Migration from Exchange Web Services to Microsoft Graph

Steps to Enable Microsoft Graph for Delegation Features:

To migrate delegated mailbox and calendar functionality from Exchange Web Services EWS to Microsoft Graph, complete the following steps:

- Upgrade Email+ to version 5.5.0 or later.
- Update the Email+ App Configuration to enable Microsoft Graph support.

> Note: Do not remove any existing EWS-related configuration keys during the transition.

- Enable the Graph API feature by adding the following value to the Key Value Pair **enabled_features** configuration:
  - graph_api

    > Before enabling the graph_api feature flag and configuring the Microsoft Graph KVPs, ensure that Email+ version 5.5.0 or later is installed on user devices. Enabling this feature on earlier versions is not supported.

- Configure the Microsoft Graph settings:
  - email_graph_host=graph.microsoft.com
  - msal_graph_resource_scope=https://graph.microsoft.com/.default
  - msg_min_allowed_auth_mode=modern_auth
  - realtime_notifications_protocol=1
- Deploy the updated configuration to managed devices through your UEM/MDM console.
- Verify delegated mailbox and calendar functionality after deployment to ensure Microsoft Graph connectivity is working as expected.

Email+ will use Microsoft Graph API for delegated mailbox, shared mailbox, delegated calendar, and shared mailbox calendar functionality.