# *Using Syncplicity with MobileIron*

September 16, 2015
Proprietary and Confidential
Do Not Distribute

## Overview

Syncplicity is a leader in Enterprise File Sync and Share (EFSS) space. Users can securely store, access and share their content using Syncplicity app. Syncplicity app provides many capabilities to IT administrators to protect data and ensure secure content sharing. Syncplicity MobileIron app for iOS will help IT administrators to securely deploy and manage their Syncplicity app on iOS devices (both iPhone and iPad). Administrators can leverage all the secure capabilities of MobileIron in this app such as prevention of data leakage, passcode enforcement and so on.

App packagename: com.syncplicity.ios.syncplicity

## App availability

Syncplicity MobileIron app for iOS is the same Syncplicity binary available in the Apple App Store. So, the app is accessible to all our customers through the App Store.

However, to make use of MobileIron secure capabilities with Syncplicity, Admins need to have the MobileIron console set up and push the this app available in App Store through the MobileIron console on the user devices. If the device is MobileIron provisioned, the Syncplicity app will work in MobileIron mode and will respect all the MobileIron AppConnect policies.

## Device compatibility

Since the MobileIron app is the same binary as our normal Syncplicity iOS app, Syncplicity MobileIron iOS App is available for iOS versions 7.0 and above.

## App-specific configuration

Syncplicity iOS App for MobileIron support these policies of MobileIron

**Data Leakage Prevention:** Administrators can prevent data leakage by configuring one or more of the following MobileIron AppConnect policies for Syncplicity iOS app for MobileIron:

- Control copy-paste only within managed AppConnect apps or in among all apps on the device.

- Control Open In only within managed AppConnect apps, among all apps on the device or within a set of Whitelisted apps.

- Wipe Content in Syncplicity: Administrators can wipe content in the Syncplicity iOS app and disable it's access if the device is lost, stolen or user's access is disabled. Administrators can wipe the content by retiring the device from their MobileIron console. In addition, they can also use Syncplicity's remote wipe policies to wipe the app and log user out.

Passcode for AppConnect apps: Administrators can protect content by enforcing passcodes for all MobileIron AppConnect apps, including Syncplicity

- Passcodes can be simple or complex and can have different number and type of required characters

- The time interval required for re-entering passcode can be configured by the admin. Passcode will required only once for any of the AppConnect apps per interval

- MobileIron's AppConnect passcode is separate from Syncplicity's passcode. Syncplicity's passcode policy is not applicable for Syncplicity iOS app for MobileIron.

Access Control: Administrators can allow the Syncplicity iOS app downloaded from MobileIron's Apps@Work app to securely connect to the enterprise servers behind firewalls by configuring AppTunneling feature of MobileIron. When AppTunneling is configured, users will be able to access Syncplicity folders behind firewall accessible only through AppTunnel.

- In addition, Administrators can prevent access to Syncplicity folders from unmanaged devices to any Syncplicity iOS app downloaded from App Store or from anywhere outside MobileIron Apps@Work(AppConnect) by configuring AppTunneling feature

# AppTunnel support

We currently support connecting to the Syncplicity iOS app using MobileIron App Tunnel. This means that in the Core Admin portal, the Syncplicity configuration should have an App Tunnel service of type "<TCP_ANY>". They should put the Sentry Host Name and Port Number.

The most useful way to use the AppTunnel feature is when your organization has an On-Premise storage solution (behind a corporate firewall) that you use Syncplicity to access. In this case, you should set the "URL Wildcard" parameter of the Syncplicity configuration to the internal IP address or domain name of the on-premise storage server. Our app communicates with its file

storage servers on a variety of ports. Thus, you must leave the "port" field empty so that connections to all ports to that IP or that domain use the AppTunnel.

## Data loss prevention policy support (iOS SDK apps only)

As mentioned in the App Specific Configuration section, copy-paste and Open In data loss prevention policies are supported in the iOS client.

- the pasteboard DLP policy
- the Open In DLP policy

## Secure file I/O support (iOS SDK apps only)

- Syncplicity does not use secure file I/O when storing its sensitive data on the device.

## AppConnect and non-AppConnect mode support (iOS SDK apps only)

Out app behaves in both AppConnect-enabled app as well as regular app because it is the same binary. The user doesn't need to do anything but if the user has MobileIron provisioned device, the app automatically works in AppConnect-enabled mode. If things change while the user is already running or AppConnect is enabled/disabled, then the user needs to quit and restart the app or best will be to logout and relogin to work in the different mode.

## User features

Users don't need any additional server or configuration to run the Syncplicity app. Once the Syncplicity app is downloaded and installed from the Mobile@Work, users can login with the Syncplicity (SSO or non-SSO) and start using the app. The app will respect all the MobileIron policies configured for the app.

## Configuration tasks

Use the following high-level steps to configure AppConnect for the app.

1. Enable AppConnect.
2. Configure an AppConnect global policy.

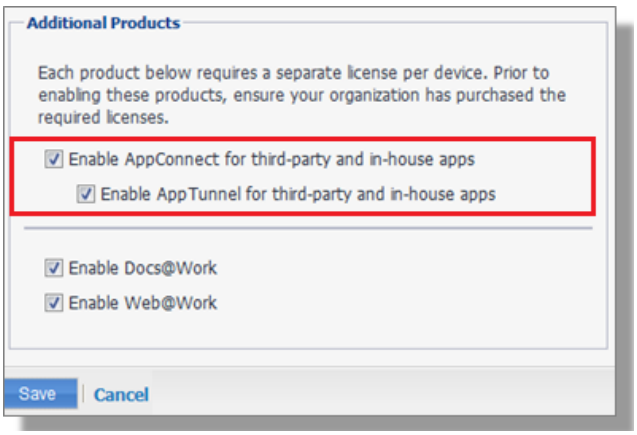
3. Configure a new AppConnect app configuration for the app.

4. Configure a new AppConnect container policy for the app.

## Enable AppConnect

Before enabling AppConnect on your Core, confirm that your organization has purchased the required AppConnect licenses. Contact your MobileIron representative if you require additional details on AppConnect license purchases.

To enable AppConnect and AppTunnel functionality on the Core, navigate to the Settings page on the Core Admin Portal and check the boxes as shown below.




1. Select the option for “Enable AppConnect for third-party and in-house apps”.

2. Select the option of “Enable AppTunnel for third-party and in-house apps”.

## Configure an AppConnect global policy

An AppConnect global policy configures the security settings for all AppConnect apps, including:

- Whether AppConnect is enabled for the devices that the policy is applied to
- AppConnect passcode requirements.

  **Note:** The AppConnect passcode is not the same as the device passcode.
- out-of-contact timeouts
- the app check-in interval

**Note:** The app check-in interval is independent of the MDM check-in timer and controls, and apps cannot be forced to check-in before the interval expires. The recommended configuration for the app check-in interval is 60 minutes.

- the default end-user message for when an app is not authorized by default
- whether AppConnect apps with no AppConnect container policy are authorized by default
- data loss prevention settings

To modify an existing AppConnect global policy:

1. On the Core Admin Portal, go to Policies & Configs > Policies.
2. Select an AppConnect global policy.
3. Click Edit.
4. Edit the AppConnect global policy based on your requirements.

   See the AppConnect and AppTunnel Guide for details about each field.

## Configure a new AppConnect app configuration

The AppConnect app configuration defines the app-specific parameters that are automatically pushed down to the app, as well as configurations for establishing and authenticating an AppTunnel associated with the app. See the AppConnect and AppTunnel Guide for details about each field.

Also, for more on AppTunnel configuration, see "Adding AppTunnel Support" in the AppConnect and AppTunnel Guide.

Use the following steps to configure the app-specific configuration:

1. On the Core Admin Portal, go to Policies & Configs > Configurations > Add New > AppConnect > App Configuration.
2. Edit the AppConnect app configuration with the Name, Description, Application, AppTunnel configuration including the identity certificate, and App-specific key-value pair configurations required for the app.

   **Note:** For the Application field, choose an application from the app distribution library, or for iOS apps, specify the iOS bundle ID. You can find the bundle ID by going to Apps > App Catalog, and clicking the hyperlink to edit the app. The bundle ID resides in the Inventory field in parenthesis.

3. AppTunnel: Click on the "Add +" button and enter the AppTunnel details. The AppTunnel service for this app must be pre-configured in order to use it here.

4. App Specific Configuration: Click on the “Add +” button to enter the key-value pair information.

## Configure a new AppConnect container policy

An AppConnect container policy specifies data loss protection policies for the app. The AppConnect container policy is required for an app to be authorized unless the AppConnect global policy allows apps without a container policy to be authorized. Such apps get their data loss protection policies from the AppConnect global policy.

Details about each field are in the AppConnect and AppTunnel Guide.

To configure an AppConnect container policy:

1. On the Core Admin Portal, go to Policies & Configs > Configurations > Add New > AppConnect > Container Policy.

2. Enter the Name, Description, and Application.

   **Note:** For the Application field, choose an application from the app distribution library, or for iOS apps, specify the iOS bundle ID. You can find the bundle ID by going to Apps > App Catalog and clicking to edit the app. The bundle ID resides in the Inventory field in parenthesis.

3. Configure the data loss protection policies according to your requirements.