

URL Scheme Replacement for iOS

Workspace ONE® data leakage prevention supports run-time replacement of the scheme in a URL that an application requests to open. This can be used, for example, to

- Cause an enterprise email client to open when the end user taps on a mailto link, instead of the system default email application.
- Cause a particular web browser to open when the end user taps on an https link, instead of the system default web browser.

URL scheme replacement is configured in the Workspace ONE Unified Endpoint Manager (UEM) console. Replacement is processed at run time by the Workspace ONE mobile Software Development Kit (SDK), if configured in the mobile application. URL scheme replacement is available in the SDK for iOS.

Table of Contents

Description.....2

Integration.....4

 Compatibility.....4

 Policy.....5

 Settings File.....6

 Console Configuration.....7

 Application Configuration.....8

 Instructions.....8

 Test Code.....11

Appendix: Decision Tree Activity Diagram.....12

Appendix: Console User Interface Screen Captures.....13

Appendix: Application Project Screen Capture.....15

Document Information.....16

Description

When an app requests to open a URL, the scheme of the URL is used by the mobile operating system to determine which other application will receive the request. Replacing the scheme will cause a different receiver to be selected.

Workspace ONE data leakage prevention supports automatic run-time replacement of the scheme in a URL (Uniform Resource Locator) value that is being opened by a mobile application that has integrated the SDK.

Workspace ONE supports replacement of these URL schemes:

- mailto
- http
- https

The feature can be configured to replace the scheme in a URL request with either:

- A built-in scheme that will select a Workspace ONE app as the receiver, Boxer for mailto, Web for http and https.
- A custom scheme configured in the UEM that will select the enterprise's chosen app as the receiver.

Workspace ONE DLP URL replacement works as follows.

- If a custom scheme is required, then a specific URL replacement policy is configured in the UEM. The policy configuration will specify which schemes are to be replaced and their replacement values. See [Policy](#) and [Settings File](#) for instructions.

This configuration isn't required for built-in scheme replacement.

- Data Leakage Prevention (DLP) is activated and configured in the UEM. DLP is a security policy. See [Console Configuration](#).
- The DLP and URL replacement policy configurations are retrieved by the SDK instance in a mobile application, at enrollment time or subsequently.
- Support for URL replacement is configured in the mobile app project, at build time. Configuration will specify which schemes are to be replaced.

The configuration is in a property list file. See [Application Configuration](#) for details.

- When the app executes an implicit or explicit request to open a URL, the SDK replaces the scheme if the following conditions are met.
 - The URL replacement configuration in the application build specifies that the scheme being opened is to be replaced.
 - DLP policy is active and specifies URL replacement.

The replacement scheme will be

- A custom scheme if there is a URL replacement configuration from the UEM, and it specifies a replacement for the scheme in the URL being opened.
 - A built-in replacement scheme otherwise.
- The request to open the URL continues to the operating system and hence to the receiver application.

See the [Decision Tree Activity Diagram](#) in the appendix to this document for a diagrammatic representation of the above.

Integration

To integrate the feature into your application, follow the instructions below.

Compatibility

Before you begin integration, ensure you have access to compatible versions of software. The following table shows the version numbers of the Workspace ONE components in which this feature first became available.

Software	Available
Workspace ONE SDK for iOS	20.6
Workspace ONE management console	9.7*

Version 9.7* is the earliest supported UEM at time of writing. All current versions of UEM have the DLP and custom settings options.

Policy

Decide on a scheme replacement policy. The policy includes these elements:

- Which of the supported schemes will be replaced.
- For each replaced scheme, which application is the required receiver.

Based on that:

- What will be the replacement scheme.

The scheme must be one registered by the required receiver application. The application developer should supply this value, for example in their product documentation.

- What name for the receiver app should be displayed to the user.

The full product name might be unfamiliar, too long, or otherwise unsuitable for display in the mobile user interface.

The policy could be recorded in a table like the following.

Original Scheme	Replacement Scheme	Replacement App Name
mailto		<i>Replacement email client name</i>
http		<i>Replacement browser name</i>
https		<i>Replacement browser name</i>

Note the following.

- Replacement of mailto is optional.
- Replacement of http and https is optional.
- The replacement app name for the http and https schemes must be the same.
- The replacement schemes for http and https needn't be the same but if http is replaced then https must also be replaced, and vice versa.

For example, the following table shows a sample policy that specifies Microsoft Outlook and the Brave browser.

Original Scheme	Replacement Scheme	Replacement App Name
mailto	ms-outlook	Outlook
http	brave	Brave
https	brave	Brave

Settings File

Create a custom settings file for insertion into your Workspace ONE management console configuration user interface. The file is a representation of the [Policy](#) decided in the preceding step.

The custom settings file will be plain text in JavaScript Object Notation (JSON) format. You can use a text editor to create the file.

The following JSON code illustrates the required structure.

```
{
  "CustomSDKSettings": {
    "com_ws1_DLP_Redirection": {
      "mailtoSchemeConfiguration": {
        "mailto": "REPLACEMENT-SCHEME-FOR-MAILTO",
        "appName": "MAILTO RECEIVER APP NAME"
      },
      "httpSchemeConfiguration": {
        "http": "REPLACEMENT-SCHEME-FOR-HTTP",
        "https": "REPLACEMENT-SCHEME-FOR-HTTPS",
        "appName": "HTTP AND HTTPS RECEIVER NAME"
      }
    }
  }
}
```

This code can also be used as a template.

- Change the capitalized parts to represent your policy.
- If your policy doesn't replace mailto, then delete the "mailtoSchemeConfiguration" entry.
- If your policy doesn't replace http and https, then delete the "httpSchemeConfiguration" entry.

Console Configuration

This feature can be configured in the Workspace ONE management console. The following instructions are intended for application developers or other users wishing to try out the feature quickly. Full documentation can be found in the online help.

These instructions assume that you have already created a [Settings File](#).

1. Log in to the management console.

The dashboard will be displayed.

2. Select an organization group.

By default, the Global group is selected.

3. Navigate to: Groups & Settings, All Settings, Apps, Settings and Policies, Security Policies.

This opens the Security Policies configuration screen, on which a number of settings can be switched on and off, and configured.

4. For the Data Loss Prevention setting, select Enabled.

When Enabled is selected, further controls will be displayed.

5. Make the following selections:

- Enable Composing Email: No

6. Select Save to commit your changes to the configuration.

7. Navigate to: Groups & Settings, All Settings, Apps, Settings and Policies, Settings.

This opens the Settings configuration screen, on which a number of settings can be switched on and off, and configured.

8. For Custom Settings, select Enabled.

When Enabled is selected, a text box, also labelled Custom Settings will be displayed.

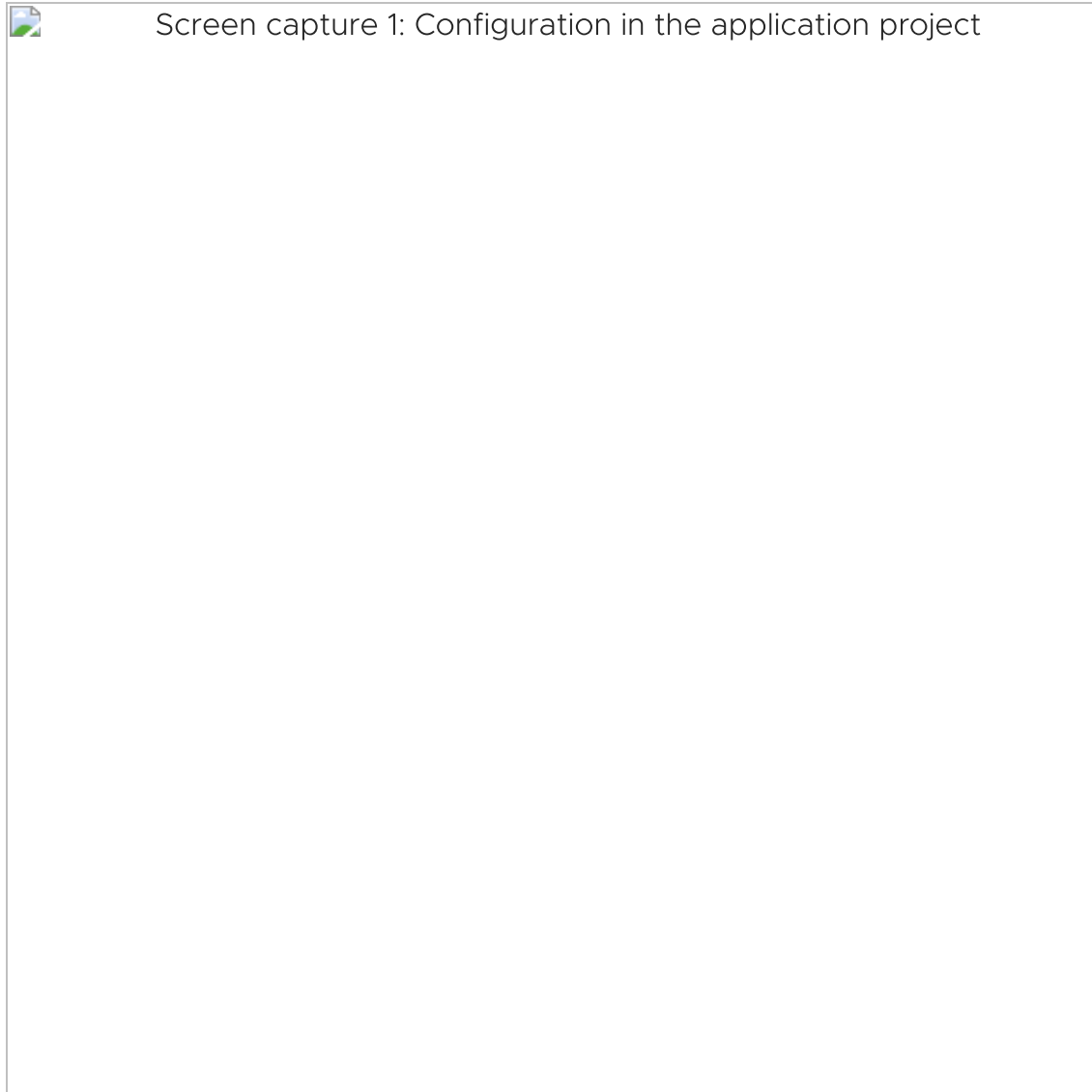
9. In the Custom Settings text box, paste in the contents of your settings file.

10. Select Save to commit your changes to the configuration.

See also the [Console User Interface Screen Captures](#) in the appendix to this document.

Application Configuration

Add a bundle and property list to your application project, as shown in the following screen capture.



Screen capture 1: Configuration in the application project

(The application name in the screen capture is Captivity, which can be ignored.)

Instructions

The following instructions will set the above configuration.

1. Open the application project in the Apple Xcode integrated development environment (IDE).

2. Add a bundle named **AWSDKDefaults** to the application, unless it already has one.

You can add a bundle like this:

1. Select the application in the project navigator.
2. In the Xcode menu, select File, New, File ...

This opens a template chooser.

3. Choose the template: Settings Bundle.

A file-save dialog open in which you will enter the name of the new bundle.

4. Enter the name: AWSDKDefaults

The full file name will be AWSDKDefaults.bundle

The AWSDKDefaults bundle has now been added and you can continue. The bundle might have some boilerplate content, like a Root property list and localisation folder. The boilerplate content can be ignored.

3. Add a property list file named **AWSDKDefaultSettings.plist** to the AWSDKDefaults bundle, unless it already has one.

You can add a property list file like this:

1. Select the application in the project navigator.
2. In the Xcode menu, select File, New, File ...

This opens a template chooser.

3. Choose the template: Property List.

A file-save dialog open in which you will enter the name of the new bundle.

4. Enter the name: AWSDKDefaultSettings

The full file name will be AWSDKDefaultSettings.plist

5. Drag and drop the new property list file into the AWSDKDefaults bundle.

The property list file has now been added in the required location and you can continue.

4. Add properties to the AWSDKDefaultSettings file.

Add dictionary properties as follows.

- To support mailto replacement, add this:

- Key: AWMailtoSchemeConfiguration
- Type: Dictionary

Inside the dictionary, add one property:

- Key: enabled
- Value: YES

- To support mailto replacement, add this:

- Key: AWURLSchemeConfiguration
- Type: Dictionary

Inside the dictionary, add one property:

- Key: enabled
- Value: YES

You can add everything by opening the property list file as source code and then pasting all or part of the following snippet.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>AWMailtoSchemeConfiguration</key>
  <dict>
    <key>enabled</key>
    <true/>
  </dict>
  <key>AWURLSchemeConfiguration</key>
  <dict>
    <key>enabled</key>
    <true/>
  </dict>
</dict>
</plist>
```

Save the property list file.

This completes the application configuration.

Test Code

Code like the following can be used to test the integration.

```
// Create a sample mailto URL.
// Assignment uses a Swift multiline string literal, see:
// https://docs.swift.org/swift-book/LanguageGuide/StringsAndCharacters.html
let mailtoString = """
mailto:\
exampleuser@example.com\
?cc=exampleuser2@example.com,exampleuser3@example.com\
&bcc=exampleuser4@example.com\
&subject=Some%20Important%20Mail\
&body=Hi%2C%0A%0A%0A
"""

if let mailtoURL = URL(string:mailtoString) {
    // Issue a request to open the URL, see:
    // https://developer.apple.com/documentation/uikit/uiapplication/1648685-open
    UIApplication.shared.open(
        mailtoURL, options: [:], completionHandler: nil)
}
```

This completes integration of URL Scheme Replacement for iOS.

Appendix: Decision Tree Activity Diagram

The following diagram represents the decision tree for Open URL with scheme replacement as a Unified Modeling Language (UML) Activity diagram.

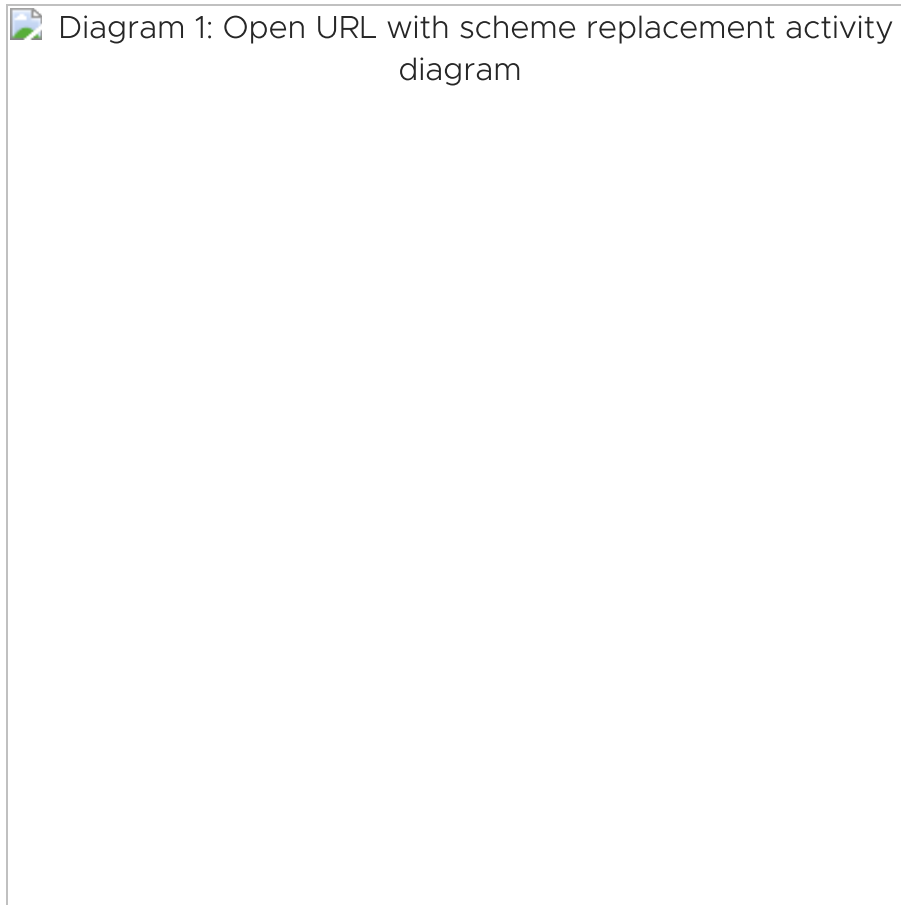


Diagram 1: Open URL with scheme replacement activity diagram

Appendix: Console User Interface Screen Captures

The following screen captures shows this feature's configuration in the management console. For step-by-step instructions, see [Console Configuration](#).



Screen capture 2: Console User Interface Data Leakage Prevention

Screen capture 2: Console User Interface Data Leakage Prevention

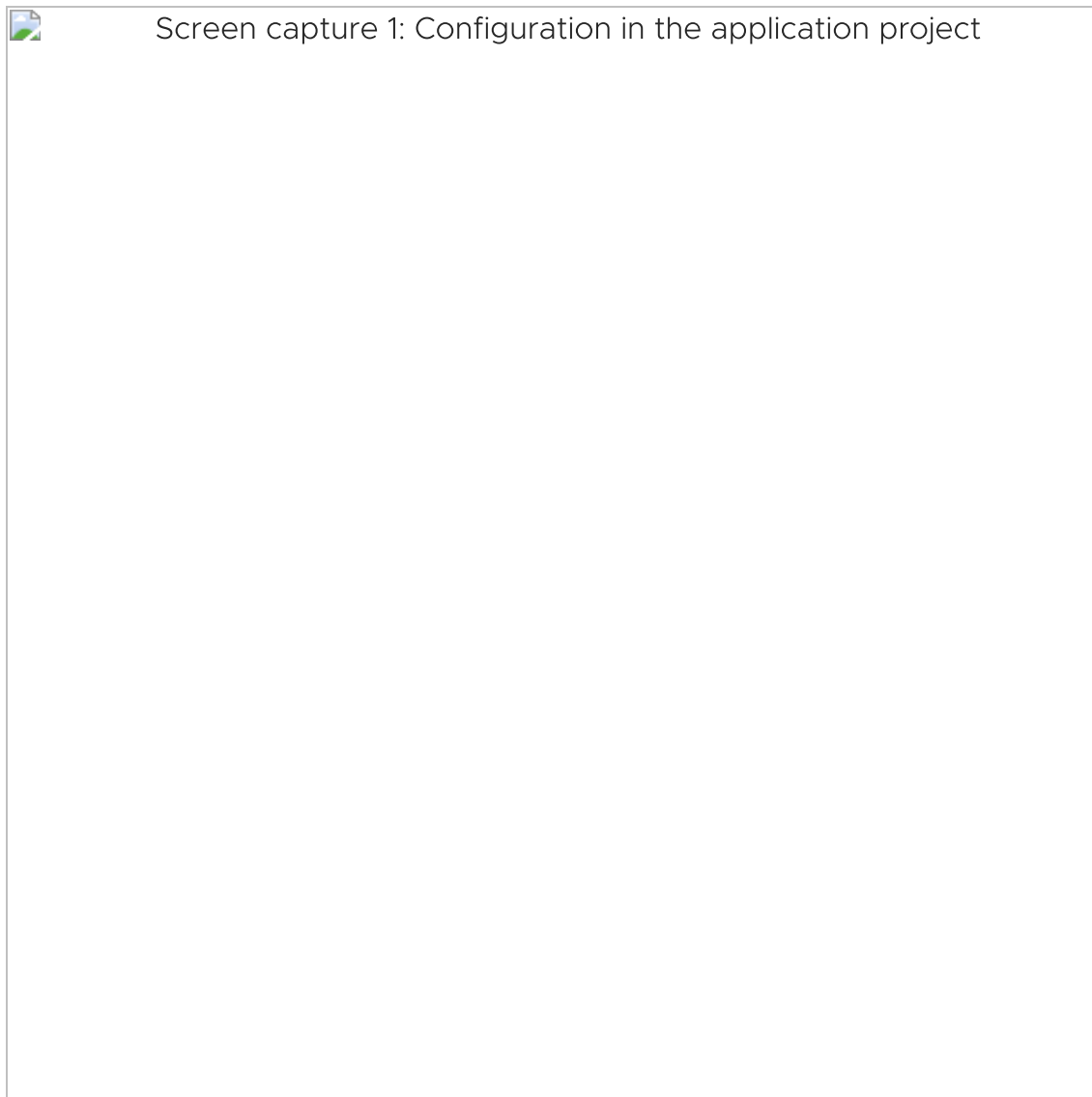


Screen capture 3: Console User Interface Custom Settings

Screen capture 3: Console User Interface Custom Settings

Appendix: Application Project Screen Capture

The following screen captures shows this feature's configuration in the Apple Xcode integrated development environment.



Screen capture 1: Configuration in the application project

(Ignore the application name in the screen capture, Captivity.)

This configuration can be set by following the [Application Configuration](#) instructions.

Document Information

Revision History

30jun2020 First Publication.

12Feb2025 Updated License

License

This software is licensed under the [Omnissa Software Development Kit \(SDK\) License Agreement](#); you may not use this software except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This software may also utilize Third-Party Open Source Software as detailed within the [open_source_licenses.txt](#) file.