Oracle® Fusion Middleware

Using the Oracle Mobile Authenticator and the Adaptive Authentication Service

11*g* Release 2 (11.1.2.3) for All Platforms

E54424-04

September 2015

ORACLE®

Using the Oracle Mobile Authenticator and the Adaptive Authentication Service, 11*g* Release 2 (11.1.2.3) for All Platforms

E54424-04

Primary Author: Michael Teger

Contributing Author: Kevin Kessler

Contributor: Vadim Lander, Vamsi Motokuru, Damien Carru, Peter Povinec, Weifang Xie, Satish Madawand, Neelima Jadhav, Charles Wesley, Harshal X Shaw, Jeremy Banford, Rey Ong, Ramana Turlapati, Deepak Ramakrishnan, David Goldsmith, Vishal Parashar

Content

Preface

This customer extract from the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management* and the *Oracle Fusion Middleware Developer's Guide for Oracle Access Management* provides administration and developer information for using the Oracle Mobile Authenticator and the Adaptive Authentication Service in an Oracle Access Management environment. This Preface covers the following topics.

Audience

This document is intended for Administrators who are familiar with:

- Oracle WebLogic Server concepts and administration
- LDAP server concepts and administration
- Database concepts and administration (for policy and session management data)
- Web server concepts and administration
- WebGate and mod_osso agents
- Auditing, logging, and monitoring concepts
- Security token concepts
- Integration of the policy store, identity store, and familiarity with Oracle Identity Management and OIS might be required

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

This Preface is for the Using the Oracle Access Management extract from the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management* and the *Oracle Fusion Middleware Developer's Guide for Oracle Access Management*. It provides details for using the Oracle Access Management implementation of the open standard OAuth 2.0 Web authorization protocol. For more information, see the following guides in the Oracle Fusion Middleware 11*g* Release 2 (11.1.2.3) documentation.

- Oracle Access Management *11g* Release 2 (11.1.2.3) *Release Notes*
- *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*—Explains how to use the Oracle Universal Installer and the WebLogic Configuration Wizard for initial Access Manager 11g deployment. Installing 11g WebGates for Access Manager is also covered.
- *Oracle Fusion Middleware Developer's Guide for Oracle Access Management*—Explains how to write custom applications and plug-ins to functions programmatically, to create custom Access Clients that protect non-Web-based resources.
- *Oracle Fusion Middleware Upgrade Guide for Java EE*—For information about the types of Java EE environments available in 10g and instructions for upgrading those environments to Oracle Fusion Middleware 11g.
- *Oracle Fusion Middleware Upgrade Guide for Oracle Identity and Access Management*
- *Oracle Fusion Middleware Migration Guide for Oracle Identity and Access Management*
- *Oracle Fusion Middleware Performance and Tuning Guide*
- *Oracle Fusion Middleware Administrator's Guide*—Describes how to manage a secure Oracle Fusion Middleware environment, including how to change ports, deploy applications, and how to back up and recover Oracle Fusion Middleware. This guide also explains how to move data from a test to a production environment.
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*—For a step-by-step guide to deployment.
- *Oracle Fusion Middleware High Availability Guide*—For high availability conceptual information as well as administration and configuration procedures for Administrators, developers, and others whose role is to deploy and manage Oracle Fusion Middleware with high availability requirements.
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference for Identity and Access Management*—Provides details on customized Identity and Access Management WLST commands.
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*—Describes how to administer and secure Web services.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introducing the Adaptive Authentication Service

The Adaptive Authentication Service offers stronger *multifactor* (also referred to as second factor) authentication for sensitive applications that require additional security in addition to the standard user name and password type authentication. Multifactor authentication involves more than one stage when verifying the identity of an entity attempting to access services from a server or on a network. For example, when multifactor authentication is enabled and configured, the traditional user name and password is the first factor. Additional security is enforced by adding a One Time Pin (OTP) step, or an Access Request (Push) Notification step as a second factor in the authentication process.

The following sections contain more details about the Adaptive Authentication Service and Access Manager Second Factor Authentication.

1.1 Using the Adaptive Authentication Service

The Adaptive Authentication Service offers the ability to add multiple steps to the authentication process. Additional security may be enforced by adding a OTP step, or an Access Request (Push) Notification step after initial user authentication. This may or may not involve the use of the Oracle Mobile Authenticator, a mobile device app that uses Time-based One Time Password and push notifications to authenticate users within the second factor authentication scheme.

> **Note:** installing Oracle Adaptive Access Manager is not required since the Adaptive Authentication Service uses a set of libraries that makes a OTP step feasible using the Oracle Mobile Authenticator.

The Adaptive Authentication Service has to be licensed and explicitly enabled in order to use it. Once the proper product license is procured you can enable the Adaptive Authentication Service using the Oracle Access Management Console. From the Oracle Access Management Console, the Adaptive Authentication Service can be enabled or disabled from the Available Services link on the Configuration Launch Pad. These section links contain more details.

- Section 3.2, "Enabling or Disabling Available Services"
- Section 2.4, "Understanding the Oracle Access Management Console"

For an introduction to the Adaptive Authentication Service and how it works, see the next section, Working with the Adaptive Authentication Service.

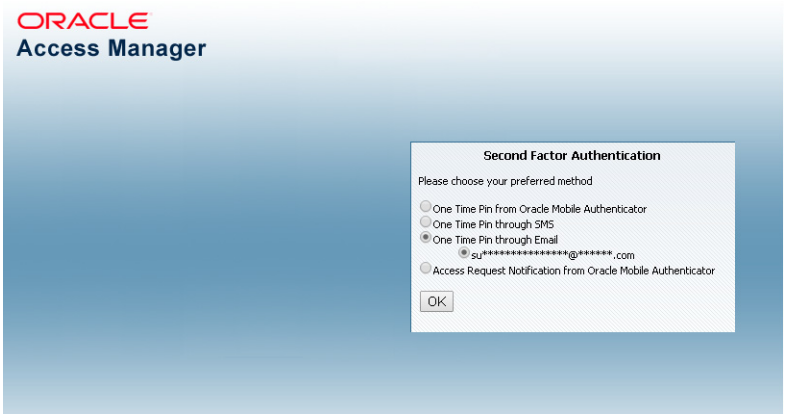
1.2 Working with the Adaptive Authentication Service

The Adaptive Authentication Service offers second factor authentication. This second factor can be a One Time Pin (OTP) or an Access Request (or push) Notification. After an initial successful user/password authentication, a Second Factor Authentication page is displayed from which the user selects their preferred method of second factor authentication. The options are:

- OTP from Oracle Mobile Authenticator
- OTP through SMS
- OTP through Email
- Access Request Notification from Oracle Mobile Authenticator

Figure 1–1 is a screenshot of the Second Factor Authentication page in which the user has selected the OTP Through Email option. In this case, the user receives the OTP via a configured Email address.

Figure 1–1 Second Factor Authentication Preferred Method Page



Screen shot of Second Factor Preferred Method Page

****

If the selected option is either OTP From Oracle Mobile Authenticator or Access Request Notification from Oracle Mobile Authenticator, the Adaptive Authentication Service works in tandem with the Oracle Mobile Authenticator (OMA), a mobile device app that uses Time-based One Time Password and push notifications to authenticate users within the second factor authentication scheme. In advance of using the OTP from OMA or Access Request Notification from OMA options, a user must download a supported authenticator app to a mobile device (for example, Oracle Mobile Authenticator to an Apple iPhone) and configure it by clicking a link provided by the Access Manager administrator. (The OMA app is not needed if using the OTP through Email or OTP through SMS options.)

Note: The Oracle Mobile Authenticator mobile device app must be configured to retrieve a secret key required to generate a OTP. Information on the secret key is in Section 1.4.1, "Generating a Secret Key for the Oracle Mobile Authenticator." Information on configuring the OMA is in Chapter 2, "Configuring the Oracle Mobile Authenticator."

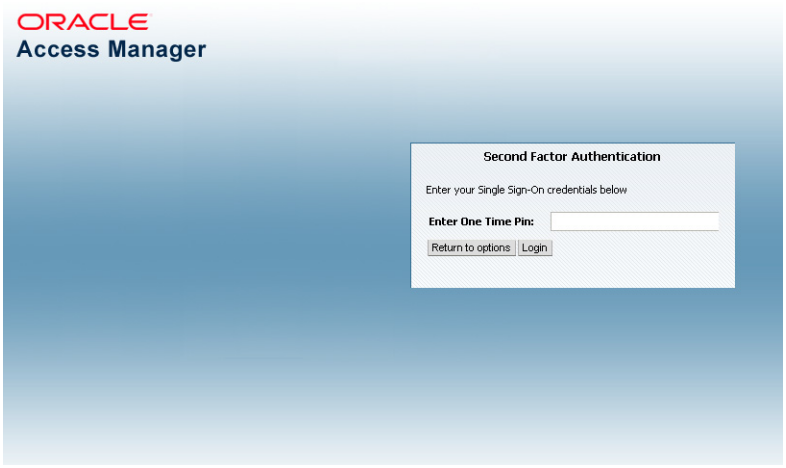
The following sections contain more details on each option and how the Oracle Mobile Authenticator works.

- Understanding the One Time Password Option
- Understanding the Access Request (Push) Notification Option
- Using the Oracle Mobile Authenticator with OTP And Access Request

1.2.1 Understanding the One Time Password Option

Let's assume the Adaptive Authentication Service is enabled and configured for second factor authentication. When the user accesses a resource protected by Access Manager, a page is displayed that requests a user name and password. If these initial credentials are authenticated successfully, a Second Factor Authentication Preferred Method Page page is displayed and the user selects from one of the options. In this use case, the user selects one of the OTP options and receives a OTP through SMS/Email or generated and displayed by the OMA app. The user enters the OTP in the OTP login page. Figure 1–2 is a screenshot of the OTP login page.

Figure 1–2 One Time Password Login Page



Screenshot of OTP Login Screen

****

Once the OTP is successfully validated by Access Manager, the user will be directed to the protected resource. On failure of any of the OTP options, an error message will be displayed, and the user will be returned to the same OTP page.

Note: Access Manager validates the OTP using the Time-based One Time Password (TOTP) algorithm. TOTP is a two-factor authentication scheme specified by the Internet Engineering Task Force (IETF) under RFC 6238 and used by the Adaptive Authentication Service. TOTP is an extension of the HMAC-based One Time Password algorithm and supports a time-based *moving factor* (a value that must be changed each time a new password is generated).

The following sections have additional details on how the user may receive the OTP.

- Using OTP through Email/SMS
- Using OTP from Oracle Mobile Authenticator

1.2.1.1 Using OTP through Email/SMS

In cases where OTP through Email or SMS is chosen, Access Manager will send a OTP to the configured email address or phone number respectively. The user then enters the received OTP and Access Manager will validate it. On a successful validation, the user will be directed to the protected resource.

The Adaptive Authentication Service expects that the required email address or phone number is configured in the appropriate field as documented in Configuring the Adaptive Authentication Plug-in. When using the OTP with Email or SMS option, the OTP is accessible from any device where the email address can be accessed or from the SMS app associated with the specified phone number, respectively.

Note: The OMA mobile app is not used for the OTP through Email or OTP through SMS options.

1.2.1.2 Using OTP from Oracle Mobile Authenticator

In the use case where a OTP will be generated and displayed by the OMA app on a mobile device, the app must be configured with the Access Manager server details. Following this configuration, the user authenticates with Access Manager using the proper credentials and Access Manager will return a secret key. This secret key is unique to each user and known only to Access Manager and the OMA. The secret key is used to generate the OTP. See Section 1.4.1, "Generating a Secret Key for the Oracle Mobile Authenticator" for information on how to populate this secret key with the required data.

After Access Manager generates a OTP for the user using the secret key, the OTP is pushed to the OMA. The user then enters the OTP in the One Time Pin Login Page. If the OTP generated by Access Manager matches the OTP entered by the user, access to the protected resource is allowed. If the OTP entries do not match, access is not allowed. See Using the Oracle Mobile Authenticator with OTP And Access Request for more details.

Note: The OMA refreshes the OTP every 30 seconds so the OTP entered by a user is valid only for that period of time.

1.2.2 Understanding the Access Request (Push) Notification Option

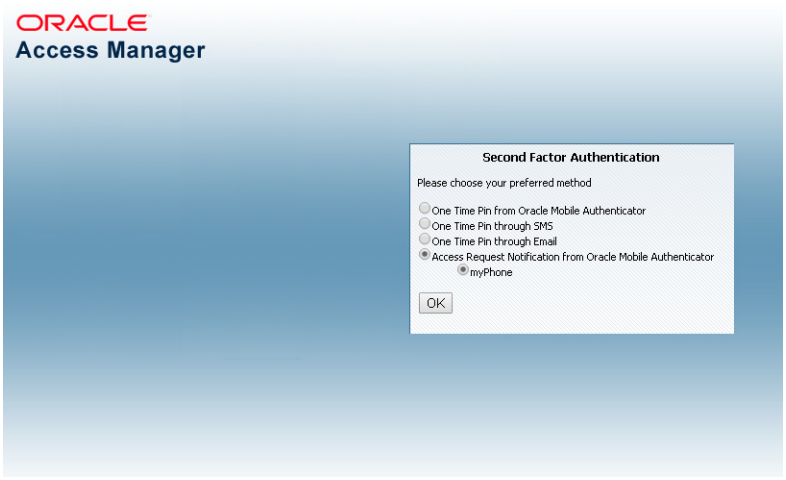
Again let's assume the Adaptive Authentication Service is enabled and configured for second factor authentication. When the user accesses a resource protected by Access

Manager, a page is displayed that requests a user name and password. If these initial credentials are authenticated successfully, a Second Factor Authentication Preferred Method Page page is displayed and the user selects from one of the options. In this use case, the user selects Access Request Notification from Oracle Mobile Authenticator.

> **Note:** This is a push notification option which works in tandem with the OMA. See Using the Oracle Mobile Authenticator with OTP And Access Request for more details.

Figure 1–3 is a screenshot of the Second Factor Authentication Preferred Method Page with Access Request Notification selected.

Figure 1–3 Access Request Notification Preferred Method Page

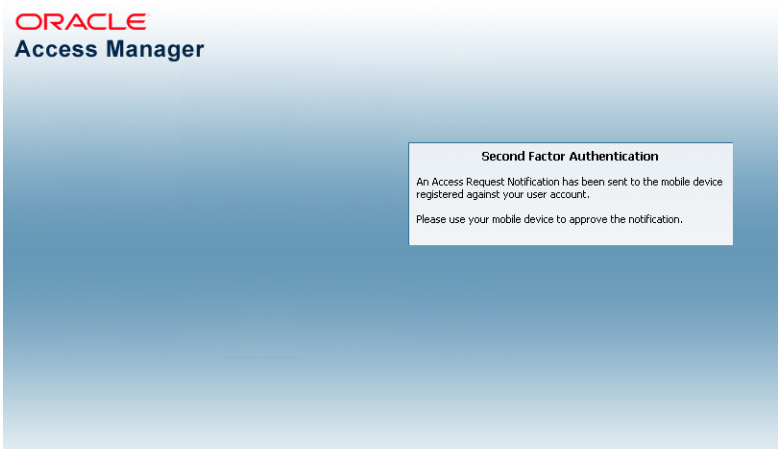


Screenshot of Preferred Method Page with Access Request Notification selected

****

When the user selects Access Request Notification from the Second Factor Authentication Preferred Method Page, Access Manager sends an Access Request Notification to either the Apple Push Notification Server or the Google Notification Server depending upon the user's configured device. The notification server then pushes a notification to the mobile device and the user will approve or deny it. Based on a successful response, the user will be directed to the protected resource. On failure, an error message will be displayed and the user will be returned to the same OTP page. Figure 1–4 is a screenshot of the Access Request Notification message displayed during this process.

Figure 1–4 Access Request Notification Wait Screen



Screenshot of Access Request Notification wait screen

****

1.2.3 Using the Oracle Mobile Authenticator with OTP And Access Request

Depending on the selected option, the Adaptive Authentication Service will need to work in tandem with the Oracle Mobile Authenticator (OMA), a mobile device app that uses Time-based One Time Password and push notifications to authenticate users with the second factor authentication scheme. To receive the OTP or Access Request Notification using the OMA, a user downloads it to an Apple or Android mobile device and configures it by clicking a link provided by the Access Manager administrator. Access Manager and OMA must share a secret key. See Section 1.4.1, "Generating a Secret Key for the Oracle Mobile Authenticator" for details on the secret key. For information on configuring OMA, see Chapter 2, "Configuring the Oracle Mobile Authenticator."

Note: The OMA app is not needed if using the OTP through Email or OTP through SMS options as discussed in Using OTP through Email/SMS.

1.3 Understanding Adaptive Authentication Service and OMA Configurations

You need to configure the Adaptive Authentication Service and, depending on the option, the OMA.

- To configure the Adaptive Authentication Service, perform the procedures documented in Section 1.4, "Configuring the Adaptive Authentication Service."
- For information on configuring the OMA, see Chapter 2, "Configuring the Oracle Mobile Authenticator."

1.4 Configuring the Adaptive Authentication Service

The following configurations are for using the Adaptive Authentication Service. It is assumed that you have installed Access Manager, a WebGate and Oracle HTTP Server

(OHS). Some of these configurations are specific to one or the other Adaptive Authentication Service options.

- Generating a Secret Key for the Oracle Mobile Authenticator
- Configuring Mobile OAuth Services to Protect the Secret Key
- Configuring the Adaptive Authentication Plug-in
- Setting Credentials for UMS, iOS and Android
- Creating a Java KeyStore for iOS Access Request (Push) Notifications
- Configuring Host Name Verifier for Android Access Request (Push) Notifications
- Configuring Access Manager for VPN Use Case

1.4.1 Generating a Secret Key for the Oracle Mobile Authenticator

A secret key needs to be shared between Access Manager and the OMA app. Businesses can generate secret keys in different ways so the means in which the secret key is generated is not important. The following RESTful endpoint is used to generate the secret key for a user in the Oracle Access Management identity store.

```
http://<HOST>:<PORT>/ms_oauth/resources/userprofile/secretkey
```

In the case of OMA online configuration (which is Oracle's recommended method of configuration), OMA uses the RESTful endpoint to store the key for a user in the identity store. In the cases of OMA manual configuration or Google Authenticator, the administrator sets up a web application which allows the user to generate a secret key also using above mentioned RESTful endpoint. The secret key is stored as a String in an LDAP attribute in the identity store and the name of this attribute must is passed to the business in the RESTful endpoint configuration before they generate the secret key. For more details, see Section 2.1, "Understanding Oracle Mobile Authenticator Configuration."

1.4.2 Configuring Mobile OAuth Services to Protect the Secret Key

Using the Oracle Access Management Console, follow this procedure to enable the Mobile and Social Service and update the User Profile Service to protect the REST Secret Key endpoint using the Basic Authentication Scheme.

1. From the Configuration Launch Pad, click Available Services.
2. Click Enable to enable Mobile and Social, if not already.
3. From the Mobile Security Launch Pad, click Mobile OAuth Services.
4. Click DefaultDomain under Mobile OAuth Identity Domains.
5. From the Resource Servers tab, click UserProfile under User Profile Services.
6. Expand the Resource URIs.
7. From the `/secretkey` tab, expand Attributes.
8. Change the value of `basicauth.allowed` to true.
9. Click Apply.

1.4.3 Configuring the Adaptive Authentication Plug-in

Access Manager provides the Adaptive Authentication Plug-in to be used for two-factor authentication. Use this procedure to configure the Adaptive Authentication Plug-in using the Oracle Access Management Console.

1. Login to the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click Authentication Plug-ins in the Plug-ins panel.
3. From the Authentication Plug-in tab, type *Adaptive* in the quick search box above the Plug-in Name column and hit Enter.

   The AdaptiveAuthenticationPlugin is displayed.
4. Change the properties displayed under Plug-in Details: AdaptiveAuthenticationPlugin as applicable to your environment.

   Table 1–1 describes the properties.

Table 1–1 Adaptive Authentication Plugin Properties

Property	Description	Default Value	Required for Challenge Method
IdentityStoreRef	Identity store name	UserIdentityStore1	All
TotpSecretKeyAttribute	Name of the user attribute in which the secret key is stored.	Attribute description	OTP using OMA, Time based OTP
TotpTimeWindow	The number of OTP codes generated by the mobile device that Access Manager will accept for validation. Since the mobile device generates a new OTP every 30 seconds, if the value is 3, Access Manager will accept the current and last three OTPs generated by the mobile device.	3	OTP using OMA, Time based OTP
PushAPNsProdServer	If set to true, the APNS production server will be used to send notifications.	false	Access Request Notifications (iOS)
PushProxyHost	Name of the proxy host if notifications are to sent to the server using a proxy.		Access Request Notifications
PushProxyPort	Proxy port if notifications are to sent to the server using a proxy.	80	Access Request Notifications
PushProxyProtocol	Proxy protocol	https://	Access Request Notifications
UmsAvailable	When Adaptive Authentication Service requires UMS to send Email and SMS, set to true.	false	SMS, Email
UmsClientUrl	URL of UMS web service		SMS, Email
PhoneField	Attribute in the identity store where the user phone number is stored	mobile	SMS

Table 1–1 (Cont.) Adaptive Authentication Plugin Properties

Property	Description	Default Value	Required for Challenge Method
EmailField	Attribute in the identity store where the user email address is stored	mail	Email
Totp_Enabled Email_Enabled Sms_Enabled Push_Enabled	Controls the options displayed in the UI. If enabled and user is not registered for Push, not setup for TOTP, or doesn't have email/phone populated in id store, those options won't be displayed. For example if user has not registered for TOTP and Push but has email populated then Email will be the only option shown.	true NOTE: Properties should be set to false only when the Administrator wants to disable a particular feature for all users.	

5. Click Save.
6. Update the same properties as applicable in the AdaptiveAuthenticationModule by clicking Authentication Modules under Plug-ins in the Access Manager Launch Pad.

   From the Authentication Modules tab, search for AdaptiveAuthenticationModule. Not all properties listed in Table 1–1 will be available.

1.4.4 Setting Credentials for UMS, iOS and Android

Use the WLST command line script to set the credentials for the Oracle User Messaging Service (UMS), the iOS certificate or the Android API key. These credentials are used by the OAM Server in the process of sending SMS/Email and push notifications. Table 1–2 contains information for the items which are needed to complete the procedure in this section.

Table 1–2 Server Side Configuration for Adaptive Authentication Service

Configuration	Information	Challenge Method
iOS Certificate/Password	https://developer.apple.com/library/mac/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html	Access Request (Push) notification using iOS
API Key	https://developers.google.com/web/updates/2015/03/push-notifications-on-the-open-web?hl=en	Access Request (Push) notification using Android
UMS Credential	UMS credentials that OAM will use to establish the connection to UMS Web service.	Email/SMS

1. `cd <MW_HOME>/oracle_common/common/bin`
2. `./wlst.sh`
3. `connect()`
4. Enter the WebLogic user name and password when prompted.

5. Press Enter to accept the default URL or modify the host and port as necessary and press Enter.
6. Run one or more of the following commands to set credentials for the UMS server, iOS or Android depending on your deployment.

> **Note:** Replace <UMS SERVER USER NAME>, <UMS SERVER PASSWORD>, <CERTIFICATE STORE PASSWORD> and <API KEY VALUE> with values specific to your environment. Do not change the values for any parameters in these commands but those listed and marked as variables.

- For OTP for email/SMS only:

```
createCred(map="OAM_CONFIG", key="umsKey", user="<UMS SERVER USER NAME>",
  password="<UMS SERVER PASSWORD>")
```

  For example:

```
createCred(map="OAM_CONFIG", key="umsKey", user="weblogic",
  password="password")
```

- For Access Request (Push) Notifications on iOS only:

```
createCred(map="OAM_CONFIG", key="pushApnsCertKey", user="apnskey",
  password="<CERTIFICATE STORE PASSWORD>")
```

  For example:

```
createCred(map="OAM_CONFIG", key="pushApnsCertKey", user="apnskey",
  password="password")
```

  See Creating a Java KeyStore for iOS Access Request (Push) Notifications for additional information when using iOS.

- For Access Request (Push) Notifications on Android only:

```
createCred(map="OAM_CONFIG", key="omaApiKey", user="omaApiKey",
  password="<API KEY VALUE>")
```

  For example:

```
createCred(map="OAM_CONFIG", key="omaApiKey", user="omaApiKey",
  password="ADDGFDGDFGRTERSDFSDFSDFTYERTERTASDASDASD")
```

7. Verify the keys by logging into Fusion Middleware Control, navigating to Domain > Security > Credentials, and checking the OAM_CONFIG map for the keys input using the commands.

> **Note:** For information on how to update, delete or otherwise manage credentials using Fusion Middleware Control, see *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*.

1.4.5 Creating a Java KeyStore for iOS Access Request (Push) Notifications

When using Access Request Notifications on iOS, create a Java KeyStore (JKS) by using the cert file and key file. Once the JKS is created, rename it as `APNsCertificate.jks` and put it in the `<domain>/config/fmwconfig` directory of the Oracle Access Management installation.

The JKS should contain the user's locally generated private key and the Apple Push Notification service (APNs) certificate downloaded from the Apple Developer Center. The following sample commands generate and import the certificate.

```
openssl x509 -in aps_production.cer -inform DER -out aps_production.pem
 -outform PEM

openssl pkcs12 -nocerts -in OMAKey.p12 -out OMAKey.pem

openssl pkcs12 -export -inkey OMAKey.pem -in aps_production.pem
 -out iOS_prod.p12

keytool -import -keystore APNsCertificate.jks -file aps_production.cer
 -alias PushCert

keytool -importkeystore -destkeystore APNsCertificate.jks
 -deststoretype JKS -srcstoretype PKCS12 -srckeystore iOS_prod.p12
```

These commands assume:

- aps_production.cer to be the name of the APNs certificate downloaded from the Apple Developer Center.
- OMAKey.p12 is the user's locally generated private key.

Also see Setting Credentials for UMS, iOS and Android.

Note: The section *Maintain Your Certificates, Identifiers, and Profiles* at the following Apple URL provides relevant information about app distribution certificates and APNs.
https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html

---

1.4.6 Configuring Host Name Verifier for Android Access Request (Push) Notifications

If you are setting up Android for Access Request notification, use the WebLogic console to update the WebLogic Managed Server for host name verification. This step is required for Access Request notification configuration on Android only. It allows the verification of host names represented using wildcards; for example, *.googleapis.com.

1. Navigate to base_domain -> Summary of Environment -> Summary of Servers -> oam_server1.
2. Click the SSL tab.
3. Expand Advanced and select the Hostname verification entry to configure the Hostname Verifier.
4. Enter `weblogic.security.utils.SSLWLSWildcardHostnameVerifier` as the Custom Hostname Verifier.
5. Click Save.
6. Restart the oam_server1.

1.4.7 Configuring Access Manager for VPN Use Case

This use case procedure illustrates how to configure Access Manager when a user will be accessing a protected resource using VPN software.

1. Login to the Oracle Access Management Console as System Administrator.
2. From the Application Security Launch Pad, click Application Domains in the Access Manager panel.

   The Application Domain tab is displayed.
3. Click Search to display all available Application Domains.
4. Click the Application Domain name that contains the resource being protected.

   The Application Domain opens in a new tab.
5. Click Authentication Policies in the Application Domain tab.
6. Click the name of the Authentication Policy that is being used to protect the particular resource for which two factor authentication is being configured.

   The appropriate Authentication Policy opens in a new tab.
7. Click Advanced Rules in the Authentication Policy tab.
8. Add a new rule by clicking the plus sign (+) under Post Authentication.

   The Add Rule dialog is displayed.
9. Enter a Rule Name and the following jython script.

   ```
   location.clientIP.startswith('10.')
   ```

   See Section 22.10.2, "Using Context Data for Advanced Rules" for details.
10. Select the AdaptiveAuthenticationScheme Authentication Scheme from the If Condition is True drop-down list.

    This Authentication Scheme will be used when the defined condition is true.
11. Click Add and then Apply to complete the procedure.

2

Configuring the Oracle Mobile Authenticator

The Oracle Mobile Authenticator is a mobile device app that uses Time-based One Time Password (TOTP) and push notifications to authenticate users with a two-factor authentication scheme. The Oracle Mobile Authenticator mobile device app must be configured to retrieve the secret key required to generate a One Time Password (OTP).

The following sections contain configuration details when using the Oracle Mobile Authenticator app on an iOS or Android mobile device.

2.1 Understanding Oracle Mobile Authenticator Configuration

The Oracle Mobile Authenticator (OMA) app can retrieve a secret key required to generate a OTP or register with Access Manager to receive push notifications. Provisioning the secret key can be done online or offline however registering for push notifications can only be done while online.

> **Note:** For details on the secret key, see Section 1.4.1, "Generating a Secret Key for the Oracle Mobile Authenticator."

- Online Configuration uses the REST web services and the Mobile OAuth Services described in Section 1.4.1, "Generating a Secret Key for the Oracle Mobile Authenticator" and Section 1.4.2, "Configuring Mobile OAuth Services to Protect the Secret Key." Once enabled, the OMA app can invoke this service to get a secret key or register for push notifications. To invoke the REST web services, OMA needs to know its location URL. In this case, the Oracle Access Management administrator creates a web page to configure the OMA. When the user taps on the web page's link (provided via e-mail), it launches the OMA, passes the location URL to the app and the REST web services location is configured. The format of the location URL is as follows.

```
oraclemobileauthenticator://settings?ServiceName::=<name_of_service>
&ServiceType::=SharedSecret/Notification/Both&
SharedSecretAuthServerType::=HTTPBasicAuthentication/OAuthAuthentication
&LoginURL::=http://<host>:<port>/secretKeyURL
&NotificationAuthServerType::= HTTPBasicAuthentication
```

```
&PushPreferencesEndpoint::=http://<host>:<port>/preferencesURL
&ChallengeAnswerEndpoint::=http://<host>:<port>/challengeAnswerURL
&SenderID::=<senderID>
&OAuthClientID::=<clientID>
&OAMOAuthServiceEndpoint::=http://<host>:<port>/oauthserviceURL
&OAuthScope::=<OAuthScope>
```

Table 2–1 documents definitions for the location URL parameters.

Table 2–1 Location URL Parameter Definitions

Parameter	Definition
ServiceName	Name of the service. This name should be unique in OMA. If another configuration with same name is sent then it will prompt the user to overwrite the previous one
ServiceType	The type of service provided by this configuration i.e. one-time password, notification or a hybrid service which combines both one-time password and notification. Value can be SharedSecret, Notification or Both.
SharedSecretAuthServerType	The type of authentication by which shared secret provisioning REST endpoint is protected. Value can be HTTPBasicAuthentication or OAuthAuthentication.
LoginURL	The REST endpoint that provisions the shared secret for generating one-time passwords. The value specified for the LoginURL query parameter is based on the OAuth settings for Oracle Mobile Authenticator.
NotificationAuthServerType	The type of authentication by which notification registration endpoint is protected. Currently only HTTP basic authentication is supported thus the value is HTTPBasicAuthentication.
PushPreferencesEndpoint	The REST endpoint where push notification preferences should be sent.
ChallengeAnswerEndpoint	The REST endpoint where push notification responses should be sent.
SenderID	The Android sender ID for sending push notifications. The SenderID is only required on Android; it is not required when using iOS.
OAuthClientID	OAuth client ID if SharedSecretAuthServerType is set for OAuth
OAMOAuthServiceEndpoint	OAM OAuth service endpoint to get OAuth profiles available on the server.
OAuthScope	The OAuth scope required to access the shared secret.

Online configuration details are also documented in Configuring the Oracle Mobile Authenticator for iOS and Configuring the Oracle Mobile Authenticator for Android. OAuth configuration details are in Chapter 2, "Configuring OAuth Services."

Note: Oracle recommends using online configuration.

- Offline Configuration supports use cases in which the mobile device can not connect to the REST end point or the parameters needed to generate the OTP are different than the defaults. The Access Manager administrator sets up a web

application which allows the user to generate or recreate a secret key. The user logs into this web application and, after authentication, the user is allowed to view the secret key and enter it in the OMA app manually. The secret key can also be delivered via an offline configuration URL so the administrator has the option of changing the OTP generation parameters (time step, hashing algorithm and the like). The format of the offline configuration URL is:

```
oraclemobileauthenticator://settings?SharedSecretValue::=<secret_key>
&AccountName::=<username>&SharedSecretEncoding::=Base32/Base64String
&OTPAlgorithm::=TOTP
&HashingAlgorithm::=MD5/SHA-1/SHA-224/SHA-256/SHA-384/SHA-512
&OTPLength::=<lenght_of_OTP>&TimeStep::=<time_in_seconds>
```

Table 2–2 contains details regarding the parameters.

Table 2–2 Offline Configuration URL Parameters

Parameter	Description
SharedSecretValue	Mandatory value is the secret key
AcountName	Prompts the user for input if omitted
SharedSecretEncoding	Default is Base32
OTPAlgorithm	Default is TOTP
Hashing Algorithm	Default is SHA-1
OTPLength	Default is 6
TimeStep	Default is 30 sec

Offline configuration details are also documented in Configuring Oracle Mobile Authenticator for Offline OTP Generation on iOS and Configuring Oracle Mobile Authenticator for Offline OTP Generation on Android.

2.2 Using the Oracle Mobile Authenticator App on iOS

The following sections contain procedures for using OMA on an iOS mobile device.

- Configuring the Oracle Mobile Authenticator for iOS
- Initializing the Oracle Mobile Authenticator on iOS
- Copying a One-Time Password from the Oracle Mobile Authenticator on iOS
- Editing an Account on the Oracle Mobile Authenticator on iOS
- Deleting an Account on the Oracle Mobile Authenticator on iOS
- Responding to Access Request (Push) Notifications on iOS
- Displaying Access Request (Push) Notifications History on iOS
- Displaying Service Account Details on iOS
- Displaying Access Manager Registered Accounts on iOS
- Displaying the OMA Version on iOS

2.2.1 Configuring the Oracle Mobile Authenticator for iOS

This procedure configures the OMA on iOS to communicate with Access Manager. A configuration URL is provided by the Access Manager administrator either by e-mail

or through a web page. Details about the URL are in Understanding Oracle Mobile Authenticator Configuration.

1. Tap the configuration URL provided by the Access Manager administrator.

   The app will open, display a unique service name to identify this app configuration, and prompt the user to accept the new settings.

2. Tap Accept to apply the settings.

   The OMA is configured to communicate with Access Manager.

2.2.2 Initializing the Oracle Mobile Authenticator on iOS

The OMA must authenticate and register an account with Access Manager. Be sure to complete Configuring the Oracle Mobile Authenticator for iOS before attempting these procedures. Any of the following procedures can be used to initialize the OMA.

- Initializing the Oracle Mobile Authenticator for OTP Generation on iOS
- Adding a OTP Generation Account Manually on iOS
- Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications Using Apple Push Notifications
- Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications and OTP Generation on iOS
- Configuring Oracle Mobile Authenticator for Offline OTP Generation on iOS

2.2.2.1 Initializing the Oracle Mobile Authenticator for OTP Generation on iOS

Once authenticated, the app receives a key from the server that will be used to generate the OTP.

1. Tap the Sign In button.

   The login screen will appear.

2. Select the OTP service name for which you are configuring second factor authentication.

   This is the unique service name defined in Configuring the Oracle Mobile Authenticator for iOS.

3. Enter your user name and password and tap Submit.

   If login is successful, you will be taken to the OTP screen for the newly added account. If login is successful but an account with the same user name for the same service already exists, you will be asked to enter a different user name. Once the user name is unique, you will be taken to the OTP screen.

2.2.2.2 Adding a OTP Generation Account Manually on iOS

You can manually configure a OTP account by entering a unique account name and key. This is the same account that would be created automatically in Initializing the Oracle Mobile Authenticator for OTP Generation on iOS.

1. Tap Enter Provided Key.

2. Enter a unique account name and key.

   If the name and key are valid, you will be taken to OTP screen for your new account. If the name is not unique or the key is not valid, you will be prompted to enter the information again.

2.2.2.3 Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications Using Apple Push Notifications

The OMA must have the user's consent to receive push notifications. It must also register successfully with the Apple Push Notification Servers and get a unique device token. Afterwards, the OMA can register with Access Manager to receive push notifications.

1. Tap the Sign In button.

   The login screen will appear.

2. Select the Push Notification service name for which you are configuring second factor authentication.

   This is the unique service name defined in Configuring the Oracle Mobile Authenticator for iOS.

3. Enter your user name and password and tap Submit.

   If authentication and registration is successful, you will be taken to the Accounts page which will display all the accounts that have been configured for Push Notifications.

2.2.2.4 Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications and OTP Generation on iOS

The OMA must have the user's consent to receive push notifications. It must also register successfully with the Apple Push Notification Servers and get a unique device token. Afterwards, the OMA can register with Access Manager to receive push notifications.

1. Tap the Sign In button.

   The login screen will appear.

2. Select the service name for which you are configuring second factor authentication.

   This is the unique service name defined in Configuring the Oracle Mobile Authenticator for iOS.

3. Enter your user name and password and tap Submit.

   If authentication and registration is successful, you will be taken to the OTP screen for the newly added account. If login is successful but a OTP account with the same user name for the same service already exists, you will be asked to enter a different user name. Once the user name is unique you will be taken to the OTP screen. Note that the newly added account will have small globe icon on the top left corner signifying that this account is also configured for push notifications.

2.2.2.5 Configuring Oracle Mobile Authenticator for Offline OTP Generation on iOS

The OMA can also be configured with a URL that contains the key used for generating a OTP. This allows for OTP generation when the mobile app is offline. This configuration URL contains the secret key so it should be delivered on a secure channel.

1. Tap on the offline configuration URL.

   This will open the OMA. If there are no OTP accounts configured with the same user name defined by the URL, the account will be added and the user will be taken to the OTP screen. If there are user name conflicts, the user will be prompted to enter a new, unique user name.

2. Enter the displayed OTP in the corresponding login page to complete authentication.

2.2.3 Copying a One-Time Password from the Oracle Mobile Authenticator on iOS

Use this procedure to copy a OTP from the OMA.

1. Tap on the account from which you want to copy the OTP.

   The Edit, Copy and Delete icons are displayed.

2. Tap the Copy icon on the left to copy the one-time password to the clipboard.
3. Paste the one-time password in the corresponding login page to complete authentication.

2.2.4 Editing an Account on the Oracle Mobile Authenticator on iOS

Use this procedure to edit an account on the OMA.

1. Tap on the account you want to edit.

   The Edit, Copy and Delete icons are displayed.

2. Tap the Edit icon in the middle to edit an account.

   A new screen in which you can edit the user name and secret key is displayed.

3. Edit the name and/or key.
4. Tap Update Account to complete the modification.

2.2.5 Deleting an Account on the Oracle Mobile Authenticator on iOS

Use this procedure to delete an account on the OMA.

1. Tap on the account you want to delete.

   The Edit, Copy and Delete icons are displayed.

2. Tap the Delete icon on the right to delete an account.

   You will be prompted for confirmation.

3. Tap Delete to confirm and delete.

2.2.6 Responding to Access Request (Push) Notifications on iOS

The OMA can receive push notifications from Access Manager if the push notification option is selected when configuring two factor authentication. An administrator can use this procedure to respond to the notifications received on the mobile device.

1. Tap the notification alert on the mobile device.

   The OMA app will come to the foreground and display notification details. This includes a user name, the resource being accessed, access time and IP address. A timer depicting how much time you have to respond to this notification is also displayed.

2. Tap Allow or Deny to control access to the resource.

   OMA will send the resource to Access Manager and remove the notification information screen.

2.2.7 Displaying Access Request (Push) Notifications History on iOS

You can see the notifications which were received by the OMA and the decision taken for that particular access request.

1. Tap on three dots icon in the top left corner.
2. Tap on Notifications button.

   All the notifications that have been received by Oracle Mobile Authenticator will be shown.
3. Tap on any of the notifications to see the details.

2.2.8 Displaying Service Account Details on iOS

You can display the services with which the OMA has been configured. This corresponds to the unique service name defined in Configuring the Oracle Mobile Authenticator for iOS.

1. Tap the three dots icon in the top left corner.
2. Tap the Configurations button.

   All the services that have been configured using this OMA will be displayed.
3. Tap a specific configuration to display the details.

   A screen will be displayed that will show all the details of the selected configuration. You can swipe from right to left to delete the configuration.

2.2.9 Displaying Access Manager Registered Accounts on iOS

You can see all the accounts that are added to the OMA and check the account type (OTP, notification or a combination of both). This corresponds to accounts configured using one of the procedures in Initializing the Oracle Mobile Authenticator on iOS.

1. Tap the three dots icon in the top left corner.
2. Tap the Accounts button.

   All the accounts that currently exist in the OMA will be displayed. Swipe from right to left to delete any account.

2.2.10 Displaying the OMA Version on iOS

You can display the version number of the OMA running on your mobile device.

1. Tap the three dots icon in the top left corner.
2. Tap the About button.

   An alert will display the OMA version.

2.3 Using the Oracle Mobile Authenticator App on Android

The following sections contain procedures for using OMA on an Android mobile device.

- Configuring the Oracle Mobile Authenticator for Android
- Initializing the Oracle Mobile Authenticator on Android
- Copying a One-Time Password from the Oracle Mobile Authenticator on Android

- Editing an Account on the Oracle Mobile Authenticator on Android
- Deleting an Account on the Oracle Mobile Authenticator on Android
- Responding to Access Request (Push) Notifications on Android
- Displaying Access Request (Push) Notifications History on Android
- Displaying Service Account Details on Android
- Displaying Access Manager Registered Accounts on Android
- Displaying the OMA Version on Android

2.3.1 Configuring the Oracle Mobile Authenticator for Android

This procedure configures the OMA on Android to communicate with Access Manager. A configuration URL is provided by the Access Manager administrator either by e-mail or through a web page. Details about the URL are in Understanding Oracle Mobile Authenticator Configuration.

1. Tap the configuration URL provided by the Access Manager administrator.

   The app will open, display a unique service name to identify this app configuration, and prompt the user to accept the new settings.

2. Tap Accept to apply the settings.

   The OMA is configured to communicate with Access Manager.

2.3.2 Initializing the Oracle Mobile Authenticator on Android

The OMA must authenticate and register an account with Access Manager. Be sure to complete Configuring the Oracle Mobile Authenticator for Android before attempting these procedures. Any of the following procedures can be used to initialize the OMA.

- Initializing the Oracle Mobile Authenticator for OTP Generation on Android
- Adding a OTP Generation Account Manually on Android
- Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications Using Google Cloud Messaging
- Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications and OTP Generation on Android
- Configuring Oracle Mobile Authenticator for Offline OTP Generation on Android

2.3.2.1 Initializing the Oracle Mobile Authenticator for OTP Generation on Android

Once authenticated, the app receives a key from the server that will be used to generate the OTP.

1. Tap the Sign In button.

   The login screen will appear.

2. Select the service name for which you are configuring second factor authentication.

   This is the unique service name defined in Configuring the Oracle Mobile Authenticator for Android.

3. Enter your user name and password and tap Submit.

If login is successful, you will be taken to the OTP screen for the newly added account. If login is successful but an account with the same user name for the same service already exists, you will be asked to enter a different user name. Once the user name is unique, you will be taken to the OTP screen.

2.3.2.2 Adding a OTP Generation Account Manually on Android

You can manually configure a OTP account by entering a unique account name and key. This is the same account that would be created automatically in Initializing the Oracle Mobile Authenticator for OTP Generation on Android.

1. Tap Enter Provided Key.
2. Enter a unique account name and key.

   If the name and key are valid, you will be taken to OTP screen for your new account. If the name is not unique or the key is not valid, you will be prompted to enter the information again.

2.3.2.3 Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications Using Google Cloud Messaging

The OMA must register successfully with the Google Cloud Messaging (Push Notification) servers and get a unique registration token. This registration token is sent to Access Manager to complete the push notification setup. Once complete, the OMA can register with Access Manager to receive push notifications.

1. Tap the Sign In button.

   The login screen will appear.
2. Select the service name for which you are configuring second factor authentication.

   This is the unique service name defined in Configuring the Oracle Mobile Authenticator for Android.
3. Enter your user name and password and tap Submit.

   If authentication and registration is successful, you will be taken to the Accounts page which will display all the accounts that have been configured for Push Notifications.

2.3.2.4 Initializing Oracle Mobile Authenticator for Access Request (Push) Notifications and OTP Generation on Android

The OMA must register successfully with the Google Cloud Messaging (Push Notification) Servers and get a unique registration token. This registration token is sent to Access Manager to complete the push notification setup. Afterwards, the OMA can register with Access Manager to receive push notifications.

1. Tap the Sign In button.

   The login screen will appear.
2. Select the service name for which you are configuring second factor authentication.

   This is the unique service name defined in Configuring the Oracle Mobile Authenticator for Android.
3. Enter your user name and password and tap Submit.

If authentication and registration is successful, you will be taken to the OTP screen for the newly added account. If login is successful but a OTP account with the same user name for the same service already exists, you will be asked to enter a different user name. Once the user name is unique you will be taken to the OTP screen. Note that the newly added account will have small globe icon on the top left corner signifying that this account is also configured for push notifications.

2.3.2.5 Configuring Oracle Mobile Authenticator for Offline OTP Generation on Android

The OMA can also be configured with a URL that contains the key used for generating a OTP. This allows for OTP generation when the mobile app is offline. This configuration URL contains the secret key so it should be delivered on a secure channel.

1. Tap on the offline configuration URL.

   This will open the OMA. If there are no OTP accounts configured with the same service name defined by the URL, the account will be added and user will be taken to the OTP screen. If there are user name conflicts, the user will be prompted to enter a new, unique service name.

2. Enter the displayed OTP in the corresponding login page to complete authentication.

2.3.3 Copying a One-Time Password from the Oracle Mobile Authenticator on Android

Use this procedure to copy a OTP from the OMA.

1. Long press on the account from which you want to copy the OTP.

   Three icons are displayed in the top/ action bar.

2. Tap the Copy icon on the left to copy the one-time password to the clipboard.
3. Paste the one-time password in the corresponding login page to complete authentication.

2.3.4 Editing an Account on the Oracle Mobile Authenticator on Android

Use this procedure to edit an account on the OMA.

1. Long press on the account you want to edit.

   Three icons are displayed in the top/ action bar.

2. Tap the Edit icon in the middle to edit an account.

   A new screen in which you can edit the user name and secret key is displayed.

3. Edit the name and/or key.
4. Tap Save to complete the modification.

2.3.5 Deleting an Account on the Oracle Mobile Authenticator on Android

Use this procedure to delete an account on the OMA.

1. Long press on the account you want to delete.

   Three icons are displayed in the top/ action bar.

2. Tap the Delete icon on the right to delete an account.

You will be prompted for confirmation.

3. Tap Delete to confirm and delete.

2.3.6 Responding to Access Request (Push) Notifications on Android

The OMA can receive push notifications from Access Manager if the push notification option is selected when configuring two factor authentication. An administrator can use this procedure to respond to the notifications received on the mobile device.

1. Tap the notification alert on the mobile device.

   The OMA app will come to the foreground and display notification details. This includes a user name, the resource being accessed, access time and IP address. A timer depicting how much time you have to respond to this notification is also displayed.

2. Tap Allow or Deny to control access to the resource.

   OMA will send the resource to Access Manager and remove the notification information screen.

2.3.7 Displaying Access Request (Push) Notifications History on Android

You can see the notifications which were received by the OMA and the decision taken for that particular access request.

1. Tap on menu in the action bar.
2. Tap on the Notifications menu item.

   All the notifications that have been received by Oracle Mobile Authenticator will be shown.

3. Tap on any of the notifications to see the details.

2.3.8 Displaying Service Account Details on Android

You can display the services with which the OMA has been configured. This corresponds to the unique service name defined in Configuring the Oracle Mobile Authenticator for Android.

1. Tap on menu in the action bar.
2. Tap the Configurations menu item.

   All the services that have been configured using this OMA will be displayed.

3. Tap a specific configuration to display the details.

   A screen will be displayed that will show all the details of the selected configuration. You can select each configuration to see the details. You can delete a configuration by selecting the delete item from the menu items in the action bar.

2.3.9 Displaying Access Manager Registered Accounts on Android

You can see all the accounts that are added to the OMA and check the account type (OTP, notification or a combination of both). This corresponds to accounts configured using one of the procedures in Initializing the Oracle Mobile Authenticator on Android.

1. Tap on the menu in the action bar.

2. Tap the Accounts menu item.

   All the accounts that currently exist in the OMA will be displayed. You can long press on an account to edit or delete it.

2.3.10 Displaying the OMA Version on Android

You can display the version number of the OMA running on your mobile device.

1. Tap the menu in the action bar.
2. Tap the About menu item to display the OMA version.

2.4 Configuring the Google Authenticator App

The Google Authenticator app only supports manual configuration. To initiate configuration in the Google Authenticator app, the user creates an account for two-factor authentication using the app. After account creation, the user manually enters the secret key received from the resource owner. (For details on the secret key, see Section 1.4.1, "Generating a Secret Key for the Oracle Mobile Authenticator.") Additionally, ensure that TOTP is enabled at the bottom of the Google Authenticator screen. Google Authenticator generates the OTP code in an offline, disconnected mode; it does not interact with Access Manager.

2.5 Using a QR Code for Configuration

A Quick Response (QR) code can be used to configure the OMA. The OMA scans the QR code for either online configuration or offline configuration details.

- In the case of online configuration, it gets the URL against which the user will be authenticated and registers the OMA app for said user. After a successful authentication and registration, the OMA gets the shared secret from the OAM server to generate the TOTP.
- In the case of offline configuration, it is assumed that the customer develops a web application and a user is authenticated by said application. The OMA scans the QR code which must have the shared secret, shared secret encoding information and optionally the OTP validity duration, the hashing algorithm to be used for TOTP or the length of the OTP (5 digits/6 digits).

The QR code needs to be created from any of the following configuration URLs.

- oraclemobileauthenticator://settings?LoginURL::=http://OAMhost:port//ms_oauth/resources/userprofile/secretkey
- oraclemobileauthenticator://settings?AuthServerType::=HTTPBasicAuthentication&&LoginURL::=http://OAMhost:port/ms_oauth/resources/userprofile/secretkey&&ServiceName::=MyBank
- oraclemobileauthenticator://settings?AuthServerType::=OAuthAuthentication&&LoginURL::=http://OAMhost:port/ms_oauth/resources/userprofile/secretkey&&ServiceName::=OAuth&&OAuthClientID::=8d91cb4821dd417286ca973045e9e25a&&OAMOAuthServiceEndpoint::=http://OAMhost:port/ms_oauth/oauth2/endpoints/oauthservice

The mobile phone user needs to go to the "Add Account" screen and select the "Scan a barcode" option. After positioning the QR code in front of the phone's camera, the OMA app will update its configuration. See "Understanding Oracle Mobile Authenticator Configuration" for additional configuration URLs.

3

Customizing Oracle Mobile Authenticator

The Oracle Mobile Authenticator is a mobile device app that uses Time-based One Time Password (TOTP) and push notifications to authenticate users. The Oracle Mobile Authenticator mobile device app is customer-facing and thus can be customized to represent your company.

This chapter describes procedures that can be used to brand the Oracle Mobile Authenticator with your company's logo and colors. It contains the following sections.

3.1 Understanding the Oracle Mobile Authenticator

The Oracle Access Management Adaptive Authentication Service offers the ability to add multiple steps to the user authentication process. This additional security may be enforced by adding a OTP step, or an Access Request (Push) Notification step after initial user authentication. In certain cases, the enforcement involves the use of the Oracle Mobile Authenticator (OMA), a mobile device app that uses Time-based One Time Password and push notifications to authenticate users within the additional second factor authentication scheme. For more details on the Adaptive Authentication Service and how it works with the OMA, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*.

3.2 Customizing Oracle Mobile Authenticator on iOS

The Oracle Mobile Authenticator (OMA) is distributed as a ZIP archive which contains OMA (as a framework), OMA resources bundle and strings files. Developers can use Xcode IDE to customize the OMA. This section contain information on how to do this. The following resources are required to customize OMA.

- `oamms_sdk_for_ios.zip` is the Identity Management Mobile SDK for iOS. It contains:
  - libIDMMobileSDK.a
  - Public Headers
  - Public Resources
- `OMACustomizable-11_1_2_3_0.zip` contains the following customizable OMA files:
  - OMALibrary.framework

- OMAResources.bundle
- Localization files

> **Note:** The `ofm_oma_clients_11.1.2.3.0.zip` contains the `OMACustomizable-11_1_2_3_0.zip` and `OracleMobileAuthenticator-11_1_2_3_0.apk` files. The latter is used in Section 3.3.1, "Using apktool."

The following sections contain more information.

- Using Xcode
- Customizing Oracle Mobile Authenticator

3.2.1 Using Xcode

The minimum version required is Xcode 6 with iOS SDK 8.0.

1. Open Xcode.
2. Click on Create a new Xcode Project.
3. Under iOS select Application.
4. Choose Single View Application and click Next.
5. Enter values for the following fields.
   - Product Name: Acme Authenticator, for example
   - Organization Name: Acme, for example
   - Organization Identifier: This value is the same as the identifier defined in Apple Developer.
   - Language: Objective-C
   - Devices: Choose Universal/iPhone/iPad depending on the devices on which this customized version of OMA will execute.
6. Click Next and then Create.

   This will open a new window where the Acme Authenticator project will be displayed.
7. In the Project Navigator menu click on Acme Authenticator project.

   The Acme Authenticator.xcodeproj tab will show the Project and Targets.
8. Under Targets click Acme Authenticator.
9. Click Build Settings.
10. Under Linking find Other Linker Flags and add -ObjC -all_load as its value.
11. Under Acme Authenticator.xcodeproj tab click General.
12. Add the following frameworks and libraries
    - Security.framework
    - SystemConfiguration.framework
    - CoreLocation.framework
    - libsqlite3.dylib

13. Under Project Navigator click Acme Authenticator and choose Add files to Acme Authenticator.
14. Add libIDMMobileSDK.a, Public Headers, Public Resources, OMALibrary.framework, OMAResources.bundle, Localization files and directories.
15. Click on AppDelegate.h file
16. Import OMALibrary app delegate by using #import <OMALibrary/OAAppDelegate.h>
17. Replace @interface AppDelegate : UIResponder <UIApplicationDelegate> with

```
@interface AppDelegate : OAAppDelegate
```

18. Click on AppDelegate.m file and remove all the UIApplicationDelegate methods.
19. Under Supporting Files right click on Info.plist file and choose Open As Source Code
20. Under the dict tag add the following tags.

```
<key>CFBundleDisplayName</key>
<string>Acme Authenticator</string>
```

21. Distribute the customized app.

    The customized Xcode project can be used for distributing the Acme Authenticator by following the guideleines in the Apple App Distribution Guide available at https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html

3.2.2 Customizing Oracle Mobile Authenticator

The following sections contain information about what can be customized.

- Changing the Application Art
- Modifying the Application Name and Text
- Toggling Online and Offline Mode
- Changing the Application Version
- Signing the Application

3.2.2.1 Changing the Application Art

Artwork used inside OMA is located in the OMAResources.bundle folder. These art files can be replaced with files of the same name. Table 3–1 contains a listing of the files. An app icon can be chosen by following the Technical Q&A QA1686 : App Icons on iPad and iPhone available at https://developer.apple.com/library/ios/qa/qa1686/_index.html

Table 3–1 Customizable Artwork

File Name	File Size	Description
check_57.fw.png	57x57 png file	Notification history screen when a notification is accepted
copy.png	57x57 png file	One-time password screen for copying OTP

Table 3–1 (Cont.) Customizable Artwork

File Name	File Size	Description
cross_57.fw.png	57x57 png file	Notification history screen when a notification was rejected
delete.png	57x57 png file	One-time password screen for deleting OTP account
edit.png	57x57 png file	One-time password screen for editing OTP account
gears_60.png	60x60 png file	Current configurations screen header
keyboard.png	57x57 png file	Add account screen and Offline configuration screen for offline account creation
notifications_57.png	57x57 png file	Notification prompt and history screen header
keyboard.png	57x57 png file	Add account screen and Online configuration screen for online account creation

3.2.2.2 Modifying the Application Name and Text

The app name can be changed by updating the value of the CFBundleDisplayName tag in the Info.plist file. The other text used in the app is pulled from the following files available under the Localization folder. This text can also be modified.

- help.html: Help file text
- privacy.html: Privacy policy text
- eula.txt: End user license agreement
- OALocalizable.strings: Messages shown in the app

3.2.2.3 Toggling Online and Offline Mode

The OMA supports both online and offline mode. This feature can be enabled or disabled by modifying the OMAResources.bundle/OAProperties.plist file.

3.2.2.4 Changing the Application Version

The Application Version can be changed by updating the CFBundleShortVersionString value in Info.plist file.

3.2.2.5 Signing the Application

App can be signed by following the instructions in the Apple App Distribution Guide available at
https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html

3.3 Customizing Oracle Mobile Authenticator on Android

The Oracle Mobile Authenticator is shipped to customers as an Android application package (.apk). The `apktool` is a tool that allows you to decompile an Android application, modify it and then rebuild it with the modifications. See the following sections for information on using the `apktool`.

- Using apktool
- Customizing Options

3.3.1 Using apktool

The `apktool` installation and usage guide can be accessed from the apktool project home at `https://code.google.com/p/android-apktool/`. The following sample command is used to decompile an Android app package.

```
apktool d "..\bin\OracleMobileAuthenticator-11_1_2_3_0.apk" -o d:\oma_smali_out
```

This next sample command is used to recompile the updated contents of Android app package. It will create a signed version of the customized app.

```
apktool b -f -a "..\Android_SDK\build-tools\20.0.0\aapt.exe"
  ..\oma_smali_out -o ..\oma_recompiled\temp.apk
```

Note: The `ofm_oma_clients_11.1.2.3.0.zip` contains the `OMACustomizable-11_1_2_3_0.zip` and `OracleMobileAuthenticator-11_1_2_3_0.apk` files. The former is used in Section 3.2, "Customizing Oracle Mobile Authenticator on iOS."

3.3.2 Customizing Options

The following sections document the customizing options for the Oracle Mobile Authenticator Android app.

- Changing Application Icons
- Modifying the Application Name and Text
- Toggling Online and Offline Mode
- Modifying the Version and Code Number
- Signing the Application

3.3.2.1 Changing Application Icons

For better UX control and multiple screen support, Android provides separate folders to better organize drawables for each screen type. (As an example the drawable-hdpi is for high pixel density devices.) Android application icons are located in the `res/` folder.

Based on the requirement the OMA application icons can also be updated in the corresponding drawable folder. In order to customize the application icons replace the old icons with the new icons without changing the icon name. Table 3–2 describes the application icons that can be customized. Again, be sure not to change the Icon name.

Table 3–2 Customizable Application Icons

Application Icon and Description	Icon Name (Do Not Modify)
App Launcher / Oracle name with padlock	ic_launcher.png
Icon to add more accounts / plus sign	add.png
Icon to initiate bar code scanning / generic barcode	barcode.png
Icon for showing notification as accepted / check mark	check.png
Icon for showing notification as canceled / x mark	cross.png
Icon for delete account / trash can	delete.png

Table 3–2 (Cont.) Customizable Application Icons

Application Icon and Description	Icon Name (Do Not Modify)
Icon for showing error alert messages / exclamation mark	error_alert.png
Icon for copy OTP (in action bar) / two paper images	ic_action_copy.png
Icon for edit account / pencil image	ic_action_edit.png
Icon to show keyboard / keyboard image	keyboard.png
Icon to show notification / globe with text balloon	notification.png
Icon for settings / generic gears image	setting.png
Icone for sign-in / generic person image	signin.png

3.3.2.2 Modifying the Application Name and Text

The name Oracle Mobile Authenticator can be customized by modifying the existing value of the string `app_name` in the `/res/values/strings.xml` file. Find the default value in the file as:

```
<string name="app_name">Oracle Mobile Authenticator</string>
```

Change this value to the preferred name and save; for example, Acme Mobile Authenticator. No special characters can be used.

```
<string name="app_name">Acme Mobile Authenticator</string>
```

The End-user License Agreement, Privacy and Help text can also be customized. To change the text, replace the original version of the file(s) with the new file(s) in the directory structure as specified below. Do not change the file name.

- End-user License Agreement: /res/raw/eula.txt
- Privacy: /res/raw/privacy.html
- Help: /res/raw/help.html

3.3.2.3 Toggling Online and Offline Mode

The Oracle Mobile Authenticator supports both online and offline mode. This feature can be enabled or disabled by modifying the /res/raw/prop.txt file. For example, to support only offline mode the content of the `prop.txt` file is defined as in Example 3–1.

Example 3–1 Customizing Oracle Mobile Authenticator Mode

```
{
"configuration":
{
"online":"no",
"offline":"yes"
}
}
```

3.3.2.4 Modifying the Version and Code Number

Modify the version and code number of the application by changing details in the `apktool.yml` located in the directory where the .apk file content has been de-compiled. (See "Using apktool.") The apktool.yml file can be viewed and modified in any text editor. The versionCode and versionName parameters are located under the

versionInfo property as illustrated in Example 3–2. In this example, the version name has been changed to test.xx.x.x from the default value 11.1.2.3.0.

Example 3–2 Changing the Android Version and Code Number

```
versionInfo:
versionCode: '3'
versionName: 'test.xx.x.x'
```

3.3.2.5 Signing the Application

Android requires that all apps be digitally signed before they can be installed. Android uses the certificate to identify the author of the app. The certificate does not need to be signed by a certificate authority so Android apps often use self-signed certificates. Additional details on this Android requirement and its process, including the procedure you can use to sign your apps, are described at http://developer.android.com/tools/publishing/app-signing.html#signing-manually