Windows Phone 8.1 Enterprise Device Management Protocol

Version: **Windows Phone 8.1 GDR2**

Last updated: **May 21, 2015**

Proprietary Notice

Contents

Windows Phone 8.1 Enterprise Device Management Protocol 1

Summary 1

Connecting to the management infrastructure (enrollment) 2
- Conceptual flow 2
- Enrollment UI 4
  - Launch workplace control panel from hyperlink 5
- Supported protocols summary 6
  - Discovery request (Steps 2–3) 6
  - Certificate enrollment policy (Steps 4–5) 6
  - Certificate enrollment (Steps 6–7) 6
  - Management configuration (Step 8) 6
- Discovery web service (Updated in Windows Phone 8.1) 6
  - Prerequisite 6
  - Description 6
  - Web Authentication Broker Support in enrollment process (New in Windows Phone 8.1) 9
  - HTTP errors 11
  - User interface 11
- Certificate enrollment policy web service 11
  - Description 11
  - SOAP faults 15
- Certificate enrollment web service 15
  - Description 15
  - Request for certificate renewal 22
  - Automatic MDM client certificate renew via Renew On Behalf Of (ROBO) function in WSTEP 24
  - Certificate renew schedule configuration 26
  - Updateability consideration 26
  - Response for certificate renewal 26
  - Configuration service providers supported during MDM enrollment and certificate renewal 27
  - SOAP faults 27
- Best practice tips 28
  - General notes 28
  - Certificates 29

Disconnecting from the management infrastructure (unenrollment) 30
- User-initiated disconnection 30
  - User unenrollment notification to the MDM server 31
- IT admin–requested disconnection 32

Enterprise settings, policies and app management 33
- DM SyncML functionality support 35
  - OMA DM standards 35
  - OMA DM protocol common elements 37
  - Device management session 38
  - OMA DM provisioning files 39

File format 40
SyncHdr element 40
Code example 41
SyncBody element 41
Code example 41
Update phone settings example 41
Code example 42
Server requirements for OMA DM 42
Enterprise OMA DM supported configuration service providers 43
DM client configuration (Updated in Windows Phone 8.1) 43
Enterprise-specific DM client configuration 44
Getting Push credentials through Windows Store 45
Acquiring application-based WNS credentials for MDM Push 45
Generate your PFN 49
Enterprise app management over DM server 50
Enterprise application install, update, uninstall (Update in Windows Phone 8.1) 50
Enterprise application restrictions (New in Windows Phone 8.1) 50
Device lock policy configuration 53
Encryption 53
Querying device encryption status 54
Enabling internal storage encryption 54
Remote wipe 54
Storage card policy configuration 54
Cellular app download limit configuration (new for GDR2) 55
Data protection under lock (new for GDR2) 55
Enterprise anti-theft override (new for GDR2) 56
Fully managed VPN setting (new for GDR2) 57
Task switcher control (new for GDR2) 57
WLAN scan frequency customization (new for GDR2) 58
Bulk enrollment (new for GDR2) 58
Apply the customization using a USB connection to the phone 58
Apply the customization using an SD card 59
Add a certificate file 59
Set the system time server 60
Set the system time zone 60
Set the language and locale 62
Set the MDM server setting 62
Sample customizations.xml 63
Certificate configuration (Updated in Windows Phone 8.1) 64
Enroll Client Certificate via Simple Certificate Enrollment Protocol 65
Enroll and manage MDM DM client certificate 69
User manually install certificates 70
Usage of user installed certificates 70
Management of user installed certificate 70
Company policy to disallow user manually install Root and CA certificates 71
Virtual Smartcard Certificate Provisioning 71
Global Certificate Revocation support 71
Phone configuration 71

Wi-Fi configuration Windows Phone 8.1 ... 71
VPN configuration Windows Phone 8.1 ... 71
Email configuration ... 72
Exchange Outlook account configuration ... 72
Internet email account configuration ... 72
Inventory cache handling ... 72

Coexistence of Exchange servers and enterprise management server ... 72

Logging support for Enterprise server creation (New in Windows Phone 8.1) ... 75
Retrieve MDM logs ... 75
View ETW logs ... 75
Steps to use WPA tool to view MDM log file ... 76

Configuration service provider reference ... 78
ActiveSync configuration service provider ... 78
Example ... 82
CertificateStore configuration service provider (Updated in Windows Phone 8.1) ... 84
Examples ... 91
DevDetail configuration service provider ... 95
DeviceLock configuration service provider ... 98
How to implement complex password requirement ... 101
Example ... 101
DevInfo configuration service provider ... 103
DMClient configuration service provider (Updated in Windows Phone 8.1) ... 104
EMAIL2 configuration service provider ... 110
Remarks ... 114
Examples ... 115
EnterpriseAppManagement configuration service provider (added functionality for Windows Phone 8.1) ... 123
Remarks ... 127
Examples ... 128
NodeCache configuration service provider ... 133
Remarks ... 134
Examples ... 135
RemoteWipe configuration service provider ... 138
Storage configuration service provider ... 138
w7 APPLICATION configuration service provider ... 139
PolicyManager configuration service provider (New in Windows Phone 8.1) ... 143
Windows Phone 8.1 supported company policies ... 145
Company Owned/Provided/Liable Device Policies ... 156
Examples ... 157
VPN configuration service provider (New in Windows Phone 8.1) ... 158
Examples ... 165
VPN single sign on configuration ... 171
WiFi configuration service provider (New in Windows Phone 8.1) ... 172
Best Practices ... 173
Examples ... 174
RemoteLock configuration service provider (New in Windows Phone 8.1) ... 178
Examples ... 180

RemoteRing configuration service provider (New in Windows Phone 8.1) 181
Examples 181
DeviceInstanceService configuration service provider 181
Examples 185
EnterpriseAssignedAccess configuration service provider (New in Windows Phone 8.1) 185
First party application Product IDs 187
Button lockdown +remap 189
Settings: System + Application settings lockdown 190
Action Center 192
Menu items 192
Start Screen size 193
Sample AssignedAccess XML 193
Sample AssignedAccess SyncML 197
Schema for AssignedAccess XML 197
Windows Embedded 8.1 Handheld device management 201
The provisioning XML file (Handheld 8.1) 201
To create a Prov.xml file to configure devices 202
Sample OMA Client Provisioning 202
Troubleshooting 205
Cryptography for prov.xml 205
Cryptographic algorithms and key lengths 206
Key management 206
Data on which crypto is applied 206
Standards and protocols 206
Crypto-related APIs 206
Set time to sync automatically over Wi-Fi (Handheld 8.1) 207
To configure NTP in OOBE 207
Enable near field communication (Handheld 8.1) 208
Components of an NFC tag and an NFC-enabled device tag 208
NFC tags are suitable for very light applications where minimal provisioning is required. The size of NFC tags that contain provisioning XML files is typically 4 KB to 10 KB. 208
To write to an NFC tag, you will need to use an NFC Writer tool, or you can use the ProximityDevice class API to write your own custom tool to transfer your provisioning XML file to your NFC tag. The tool must publish a binary message (write) a Chunk data type to your NFC tag 208
The following table describes the information that is required when writing to an NFC tag 208
NFC-enabled device tag components 209
Enable or disable NFC capabilities 210
EnableDeviceEnrollment Request and Response (Handheld 8.1) 211
Modified certificate enrollment web service request 211
SOAP Request 211
X509 certificate request 213
SOAP response 213
Apps Corner (Handheld 8.1) 216
Assigned Access (Handheld 8.1) 216
EnterpriseAssignedAccess configuration service provider (Handheld 8.1) 216
OMA client provisioning examples 228
OMA DM examples 229

EnterpriseExt configuration service provider (Handheld 8.1) 233
The restart process 238
The enrollment process 238
OMA client provisioning examples 238
OMA DM examples 238
Schema for the maintenance ScheduleXML parameters 241
EnterpriseExtFileSystem configuration service provider (Handheld 8.1) 243
OMA DM examples 246
Reference 247
Q&A 247
Appendix 250
XSD for ApplicationRestriction policy in PolicyManager 250
XML samples for ApplicationRestriction policy in PolicyManager 252
Known Issues 264
Support 264

Summary

Windows Phone 8 provides an enterprise management solution to help IT pros manage company security policies and business applications, while avoiding compromise of the users' privacy on their personal phones.

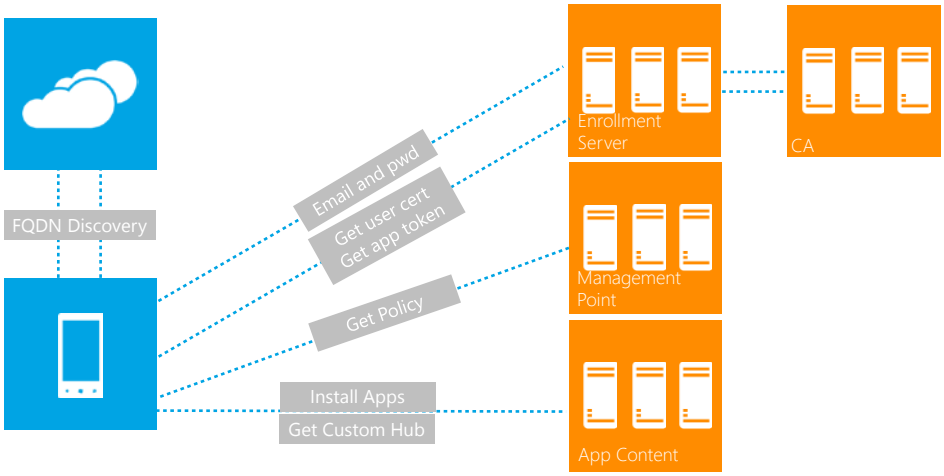
A built-in management component can communicate with the management server. There are two parts to the Windows Phone 8.1 management component:

- The enrollment client, which enrolls and configures the phone to communicate with the enterprise management server.
- The phone management client, which periodically synchronizes with the management server to check for updates and apply the latest policies set by IT.

A custom Windows Phone app that can be downloaded during enrollment is an optional component to create an end-to-end phone management experience. This custom app can be created such that the user can discover and install available LOB apps from an enterprise app catalog. This app can be created by an enterprise as a custom Company Hub or by the Mobile Device Management (MDM) vendor. For more information about Company Hub apps, see Developing a Company Hub app on MSDN.

Third-party MDM servers can manage Windows Phone 8.1 by using the Enterprise Device Management protocol. The built-in management client is able to communicate with a third-party server proxy that supports the protocols outlined in this document to perform enterprise management tasks. The third-party server will have the same consistent first-party user experience for enrollment, which also provides simplicity for Windows Phone users. MDM servers do not need to create or download a client to manage Windows Phone.

The following diagram shows the overall Enterprise Device Management architecture.



Enterprise Device Management Architecture

Connecting to the management infrastructure (enrollment)

The first thing to enable enterprise management is to configure the phone to communicate with the MDM server using security precautions. This is done via the enrollment process described in this section. The enrollment service verifies that only authenticated and authorized phones can be configured to be managed by their enterprise.

The enrollment process includes four steps:

1. Discovery of the enrollment endpoint

   This step provides the enrollment endpoint configuration settings.

2. Certificate installation

   This step handles user authentication, certificate generation, and certificate installation. The installed certificates will be used in the future to manage client/server SSL mutual authentication.

3. Enterprise application token and first app installation (recommended is an app that allows the user to discover more enterprise apps, such as a Company Hub).

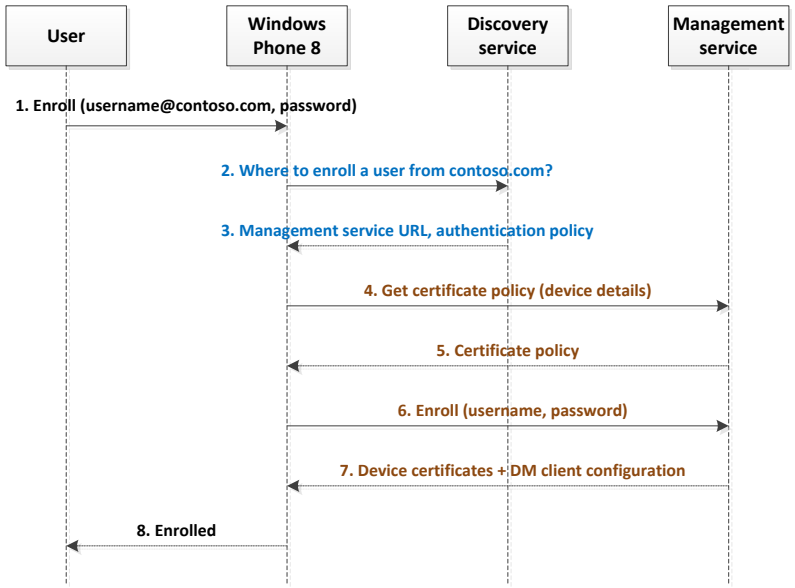
   This step installs the enterprise application token that allows users to download and install enterprise applications. A Company Hub application could be installed at the end of the enrollment process to allow users to easily find out what business applications are available.

4. DM client provisioning

   This step configures the DM client to connect to a Mobile Device Management (MDM) server after enrollment via DM SyncML over HTTPS (aka OMA DM XML).

Conceptual flow

The following diagram illustrates the enrollment flow. The following examples refer to the fictional company Contoso, whose website is contoso.com.

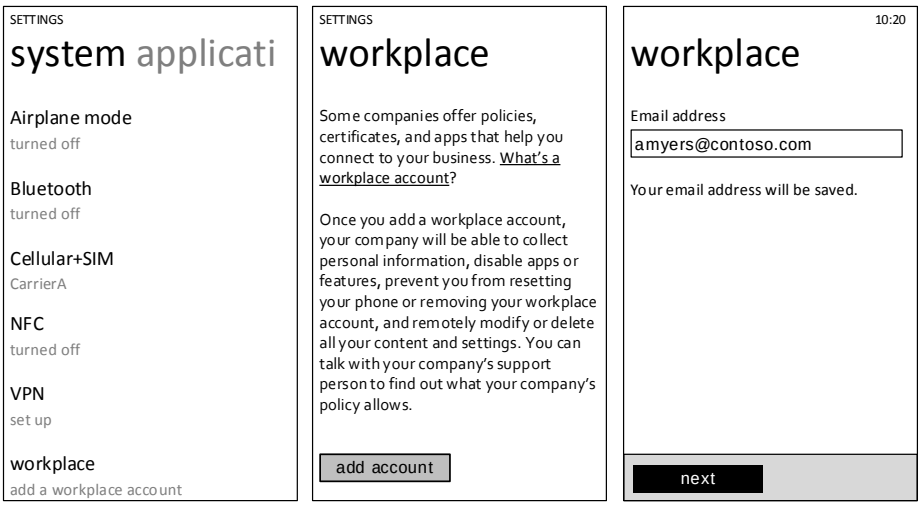


Enrollment Flow

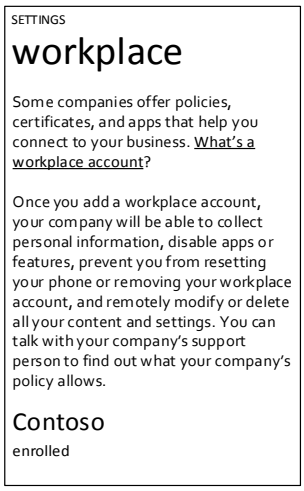
Step	Description
1	The user clicks on the enrollment application and inserts their credentials.
2	The phone sends a discover request to the discovery service. The request includes the domain part of the email address.
3	The discovery service returns a response that contains the management service URL and optionally an authentication policy. The authentication policy includes information on the authentication method and required authentication steps.
4, 5	The phone contacts the management service and asks for a certificate policy. The management service returns a certificate policy. The returned certificate issuers provide an X.509 v3 security token by using MS-WSTEP.
6, 7	Based on the authentication policy data (step 3) and the certificate policy, the phone creates an enrollment request. The management service generates a phone certificate and provides DM client settings. The phone installs the certificate, an enterprise application token (optional), a link to an enterprise application (optional), and initiates a DM request to the MDM server.
8	The phone notifies the user that enrollment is finished.

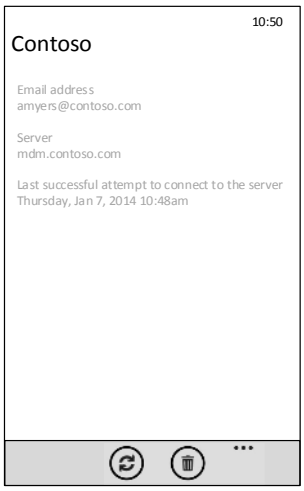
Enrollment UI

The following mockup shows the user experience of enrollment. Notice that this user experience is the enrollment client's built-in UI; there is no third-party extensibility unless the server supports web authentication broker for authentication during enrollment. The user enters a corporate email address. The phone tries to auto-discover the server and start the enrollment process. If no discovery enrollment server is found, the phone presents a screen to allow the user to enter the server address. Depending on which authentication is supported by the server, the user will be presented to enter some requested credential. Once enrollment is complete, a workplace account will be added to the workplace setting control panel.



If federated WAB based authentication is used, a server authentication page is displayed.





If auto discovery fails, the user is given the option to manually enter discovery server address.

Launch workplace control panel from hyperlink

Windows Phone 8.1 supports the launching of the workplace control panel using a hyperlink: ms-settings-workplace, formatted to Settings Workplace.

The enterprise can leverage this functionality to build a better user experience , for example, by instructing the user to create a workplace account using a workplace setup email. The email can include above hyperlink for the user to click directly to launch workplace control panel to start enrolling.

Supported protocols summary

The following subsections describe the protocols that are used in the enrollment flow.

Discovery request (Steps 2–3)

The discovery request is a simple HTTP post call that returns XML over HTTP. The returned XML includes the authentication URL, the management service URL, and the user credential type.

Certificate enrollment policy (Steps 4–5)

The certificate enrollment policy configuration is an implementation of the MS-XCEP protocol, which is described in [MS-XCEP]: X.509 Certificate Enrollment Policy Protocol Specification. Section 4 of the specification provides an example of the policy request and response. The X.509 Certificate Enrollment Policy Protocol is a minimal messaging protocol that includes a single client request message (GetPolicies) with a matching server response message (GetPoliciesResponse).

Certificate enrollment (Steps 6–7)

The certificate enrollment is an implementation of the MS-WSTEP protocol.

Management configuration (Step 8)

The server sends provisioning XML that contains a server certificate (for SSL server authentication), a client certificate issued by enterprise CA, DM client bootstrap information (for the client to communicate with the management server), an enterprise application token (for the user to install enterprise applications), and the link to download the Company Hub application.

Discovery web service (Updated in Windows Phone 8.1)

Prerequisite

The administrator of the discovery service must create a host with the address enterpriseenrollment.*domain*.com.

Description

The discovery web service provides the configuration information necessary for a user to enroll a phone with a management service. The service is a restful web service over HTTPS (server authentication only).

Request:

The phone's automatic discovery flow uses the domain name of the email address that was submitted to the Workplace settings screen during sign in. The automatic discovery system constructs a URI that uses this hostname by appending the subdomain "enterpriseenrollment" to the domain of the email address, and by appending the path "/EnrollmentServer/Discovery.svc". For example, if the email address is "sample@contoso.com", the resulting URI for first Get request would be: http://enterpriseenrollment.contoso.com/EnrollmentServer/Discovery.svc

The first request is a standard HTTP GET request.

The following example shows a request via HTTP GET to the discovery server given `user@contoso.com` as the email address.

Request:

```
Request Full Url: http://EnterpriseEnrollment.contoso.com/EnrollmentServer/Discovery.svc
Content Type: unknown
Header Byte Count: 153
Body Byte Count: 0
```

Header:

```
GET /EnrollmentServer/Discovery.svc HTTP/1.1
User-Agent: Windows Phone 8 Enrollment Client
Host: EnterpriseEnrollment.contoso.com
Pragma: no-cache
```

Response:

```
Request Full Url: http://EnterpriseEnrollment.contoso.com/EnrollmentServer/Discovery.svc
Content Type: text/html
Header Byte Count: 248
Body Byte Count: 0
```

Header:

```
HTTP/1.1 200 OK
Connection: Keep-Alive
Pragma: no-cache
Cache-Control: no-cache
Content-Type: text/html
Content-Length: 0
```

After the phone gets a response from the server, the phone sends a POST request to enterpriseenrollment.*domain.name*/EnrollmentServer/Discovery.svc. After it gets another response from the server (which should tell the phone where the enrollment server is), the next message sent from the phone is to enterpriseenrollment.*domain.name* to the enrollment server.

The following logic is applied:

1. The phone first tries HTTPS. If the server cert is not trusted by the phone, the HTTPS fails.
2. If that fails, the phone tries HTTP to see whether it is redirected:
    - If the phone is not redirected, it prompts the user for the server address.
    - If the phone is redirected, it prompts the user to allow the redirect.

The following example shows a request via an HTTP POST command to the discovery web service given `user@contoso.com` as the email address,

```
https://EnterpriseEnrollment.Contoso.com/EnrollmentServer/Discovery.svc
```

Header:

```
POST /EnrollmentServer/Discovery.svc HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
User-Agent: Windows Phone 8 Enrollment Client
Host: EnterpriseEnrollment.Contoso.com
Content-Length: xxx
Cache-Control: no-cache
```

```
<?xml version="1.0"?>
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
   xmlns:s="http://www.w3.org/2003/05/soap-envelope">
   <s:Header>
      <a:Action s:mustUnderstand="1">
 http://schemas.microsoft.com/windows/management/2012/01/enrollment/IDiscoveryService/Discover
      </a:Action>
      <a:MessageID>urn:uuid: 748132ec-a575-4329-b01b-6171a9cf8478</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://ENROLLTEST.CONTOSO.COM/EnrollmentServer/Discovery.svc
      </a:To>
   </s:Header>
   <s:Body>
      <Discover xmlns="http://schemas.microsoft.com/windows/management/2012/01/enrollment/">
         <request xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <EmailAddress>user@contoso.com</EmailAddress>
            <RequestVersion>2.0</RequestVersion> <!-- Updated Windows Phone 8.1 -->
             <DeviceType>WindowsPhone</DeviceType> <!--Added in Windows Phone 8.1-->
         </request>
      </Discover>
   </s:Body>
</s:Envelope>
```

Response

The discovery response is in the XML format and includes the following fields:

- Enrollment service URL (EnrollmentServiceUrl) – Specifies the URL of the enrollment endpoint that is exposed by the management service. The phone should call this URL after the user has been authenticated. This field is mandatory.
- Authentication policy (AuthPolicy) – Indicates what type of authentication is required. For the MDM server, OnPremise is the supported value, which means that the user will be authenticated when calling the management service URL. This field is mandatory.
- In Windows Phone 8.1, Federated is added as another supported value. This allows the server to leverage the Web Authentication Broker to perform customized user authentication, and term of usage acceptance.
-

Note that the HTTP server response must not be chunked; it must be sent as one message.

The following example shows a response received from the discovery web service for OnPremise authentication:

Header:

```
HTTP/1.1 200 OK
Content-Length: 865
Content-Type: application/soap+xml; charset=utf-8
Server: EnterpriseEnrollment.Contoso.com
Date: Tue, 02 Aug 2012 00:32:56 GMT
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action s:mustUnderstand="1">
http://schemas.microsoft.com/windows/management/2012/01/enrollment/IDiscoveryService/DiscoverR
esponse
```

```
      </a:Action>
      <ActivityId>
         d9eb2fdd-e38a-46ee-bd93-aea9dc86a3b8
      </ActivityId>
      <a:RelatesTo>urn:uuid: 748132ec-a575-4329-b01b-6171a9cf8478</a:RelatesTo>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <DiscoverResponse
         xmlns="http://schemas.microsoft.com/windows/management/2012/01/enrollment">
         <DiscoverResult>
            <AuthPolicy>OnPremise</AuthPolicy>
            <EnrollmentPolicyServiceUrl>
               https://enrolltest.contoso.com/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
            </EnrollmentPolicyServiceUrl>
            <EnrollmentServiceUrl>
               https://enrolltest.contoso.com/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
            </EnrollmentServiceUrl>
         </DiscoverResult>
      </DiscoverResponse>
   </s:Body>
</s:Envelope>
```

Web Authentication Broker Support in enrollment process (New in Windows Phone 8.1)

Windows Phone 8.1 adds the support of a Federated as supported AuthPolicy value. When authentication policy is set to be Federated, Web Authentication Broker (WAB) will be leveraged by the enrollment client to get a security token. The WAB start page URL is provided by the discovery service in the response message. The enrollment client will call the WAB API within the response message to start the WAB process. WAB pages are server hosted web pages. The server should build those pages to fit the phone screen nicely and be as consistent as possible to other builds in the MDM enrollment UI. The opaque security token that is returned from WAB as an endpage will be used by the enrollment client as the device security secret during the client certificate enrollment request call.

A new XML tag, AuthenticationServiceUrl, is introduced in the DiscoveryResponse XML to allow the server to specify the WAB page start URL. For Federated authenticaitonauthentication, this XML tag must exist. For OnPremise authentication, this XML tag must not exist.

Note that the enrollment client is agnostic with regards to the protocol flows for authenticating and returning the security token. While the server might prompt for user credentials directly or enter into a federation protocol with another server and directory service, the enrollment client is agnostic to all of this. To remain agnostic, all protocol flows pertaining to authentication that involve the enrollment client are passive, that is, browser-implemented.

The following are the explicit requirements for the server.

- The <DiscoveryResponse><AuthenticationServiceUrl> element MUST support HTTPS.
- The auth sever must use a device trusted root certificate. Otherwise, the WAP call will fail.
- WP doesn't support Window Integrated Authentication (WIA) for ADFS during WAB authentication. ADFS 2012 R2 if used needs to be configured to not attempt WIA for Windows Phone 8.1 device.

The enrollment client issues an HTTPS request as follows:

```
AuthenticationServiceUrl?appru=<appid>&login_hint=<User Principal Name>
```

- <appid> is of the form ms-app://string
- <User Principal Name> is the name of the enrolling user, for example, user@constoso.com as inputted by the user in an enrollment sign in page. The value of this attribute serves as a hint that can be used by the authentication server as part of the authentication.

After authentication is complete, the auth server SHOULD return an HTML form document with a POST method action of *appid* identified in the query string parameter. For example:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Content-Length: 556

<!DOCTYPE>
<html>
   <head>
      <title>Working...</title>
      <script>
         function formSubmit() {
            document.forms[0].submit();
         }
           window.onload=formSubmit;
      </script>
   </head>
   <body>
    <!-- appid below in post command must be same as appid in previous client https request. -
->
      <form method="post" action="ms-app://appid">
         <p><input type="hidden" name="wresult" value="token value"/></p>
         <input type="submit"/>
      </form>
   </body>
</html>
```

The server has to send a POST to a redirect URL of the form ms-app://string (the URL scheme is ms-app) as indicated in the POST method action. The security token value is the base64-encoded string "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd#base64binary" contained in the <wsse:BinarySecurityToken> EncodingType attribute. Windows Phone 8.1 does the binary encode when it sends it back to enrollment server, in the form it is just HTML encoded. This string is opaque to the enrollment client; the client does not interpret the string.

The following example shows a response received from the discovery web service which requires authentication via WAB:

Header:

```
HTTP/1.1 200 OK
Content-Length: 865
Content-Type: application/soap+xml; charset=utf-8
Server: EnterpriseEnrollment.Contoso.com
Date: Tue, 02 Aug 2012 00:32:56 GMT
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action s:mustUnderstand="1">
http://schemas.microsoft.com/windows/management/2012/01/enrollment/IDiscoveryService/DiscoverR
esponse
      </a:Action>
      <ActivityId>
         d9eb2fdd-e38a-46ee-bd93-aea9dc86a3b8
```

```
      </ActivityId>
      <a:RelatesTo>urn:uuid: 748132ec-a575-4329-b01b-6171a9cf8478</a:RelatesTo>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <DiscoverResponse
         xmlns="http://schemas.microsoft.com/windows/management/2012/01/enrollment">
         <DiscoverResult>
            <AuthPolicy>Federated</AuthPolicy>
            <EnrollmentPolicyServiceUrl>
               https://enrolltest.contoso.com/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
            </EnrollmentPolicyServiceUrl>
            <EnrollmentServiceUrl>
               https://enrolltest.contoso.com/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
            </EnrollmentServiceUrl>
            <AuthenticationServiceUrl>
               https://portal.manage.contoso.com/LoginRedirect.aspx
            </AuthenticationServiceUrl>
         </DiscoverResult>
      </DiscoverResponse>
   </s:Body>
</s:Envelope>
```

HTTP errors

The following list specifies the possible faults:

- Redirection 3xx.
- 404 – No CNAME DNS record was registered.
- Server error 5xx.

User interface

The phone provides the user with the option to enter the discovery URL manually if automatic discovery fails. This could be used to support scenarios such as:

- An enterprise wants to migrate from one management service to another gradually, and as part of the migration, it wants to let only selected people enroll with the new service. Those people will have to enter the discovery URL manually in order to enroll with the new service.
- The discovery URL cannot be constructed from the email address of the user. For example, Contoso sets up https://discovery.managementservice.contoso.com, but some employees' email addresses are in a different domain (e.g., *user*@europe.contoso.com).

Certificate enrollment policy web service

Description

Policy service is optional. By default, if no policies are specified, the minimum key length is 2k and the hash algorithm is SHA-1.

This web service implements the X.509 Certificate Enrollment Policy Protocol (MS-XCEP) specification that allows customizing certificate enrollment to match different security needs of enterprises at different times (cryptographic agility). The service processes the GetPolicies message from the client, authenticates the client, and returns matching enrollment policies in the GetPoliciesResponse message.

Request

For the OnPremise authentication policy, the `UsernameToken` in `GetPolicies` contains the user credential, whose value is based on the authentication policy in discovery. A sample of the request can be found on the MSDN website; the following is another sample, with `user@contoso.com` as the user name and `mypassword` as the password.

Header:

```
POST /ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
User-Agent: Windows Phone 8 Enrollment Client
Host: enrolltest.contoso.com
Content-Length: xxxx
Cache-Control: no-cache
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing"
   xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"
   xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
   xmlns:ac="http://schemas.xmlsoap.org/ws/2006/12/authorization">
   <s:Header>
      <a:Action s:mustUnderstand="1">
         http://schemas.microsoft.com/windows/pki/2009/01/enrollmentpolicy/IPolicy/GetPolicies
      </a:Action>
      <a:MessageID>urn:uuid:72048B64-0F19-448F-8C2E-B4C661860AA0</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://enrolltest.contoso.com/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
      </a:To>
      <wsse:Security s:mustUnderstand="1">
         <wsse:UsernameToken u:Id="uuid-cc1ccc1f-2fba-4bcf-b063-ffc0cac77917-4">
            <wsse:Username>user@contoso.com</wsse:Username>
            <wsse:Password wsse:Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">mypassword</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <GetPolicies
         xmlns="http://schemas.microsoft.com/windows/pki/2009/01/enrollmentpolicy">
         <client>
            <lastUpdate xsi:nil="true"/>
            <preferredLanguage xsi:nil="true"/>
         </client>
         <requestFilter xsi:nil="true"/>
      </GetPolicies>
   </s:Body>
</s:Envelope>
```

For Federated authentication policy, The security token credential is provided in a request message using the <wsse:BinarySecurityToken> element [WSS]. The security token is retrieved as described in the discovery response section. The authentication information is as follows:

wsse:Security: The enrollment client implements the <wsse:Security> element defined in [WSS] section 5. The <wsse:Security> element MUST be a child of the <s:Header> element.

wsse:BinarySecurityToken: The enrollment client implements the <wsse:BinarySecurityToken> element defined in [WSS] section 6.3. The <wsse:BinarySecurityToken> element MUST be included as a child of the <wsse:Security> element in the SOAP header.

As was described in the discovery response section, the inclusion of the <wsse:BinarySecurityToken> element is opaque to the enrollment client, and the client does not interpret the string, and the inclusion of the element is agreed upon by the security token authentication server (as identified in the <AuthenticationServiceUrl> element of <DiscoveryResponse> and the enterprise server.

The <wsse:BinarySecurityToken> element contains a base64-encoded string. The enrollment client uses the security token received from the authentication server and base64-encodes the token to populate the <wsse:BinarySecurityToken> element.

wsse:BinarySecurityToken/attributes/ValueType: The <wsse:BinarySecurityToken> ValueType attribute MUST be "http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentUserToken".

wsse:BinarySecurityToken/attributes/EncodingType: The <wsse:BinarySecurityToken> EncodingType attribute MUST be "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd#base64binary".

The following is an enrollment policy request example with a received security token as client credential.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing"
   xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"
   xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
   xmlns:ac="http://schemas.xmlsoap.org/ws/2006/12/authorization">
   <s:Header>
      <a:Action s:mustUnderstand="1">
         http://schemas.microsoft.com/windows/pki/2009/01/enrollmentpolicy/IPolicy/GetPolicies
      </a:Action>
      <a:MessageID>urn:uuid:72048B64-0F19-448F-8C2E-B4C661860AA0</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://enrolltest.contoso.com/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
      </a:To>
      <wsse:Security s:mustUnderstand="1">
         <wsse:BinarySecurityToken  wsse:ValueType=
http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentUserToken
                  wsse:EncodingType=
   http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd#base64binary"
                  xmlns=
          "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                  B64EncodedSampleBinarySecurityToken
               </wsse:BinarySecurityToken>
      </wsse:Security>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <GetPolicies
```

```
         xmlns="http://schemas.microsoft.com/windows/pki/2009/01/enrollmentpolicy">
         <client>
            <lastUpdate xsi:nil="true"/>
            <preferredLanguage xsi:nil="true"/>
         </client>
         <requestFilter xsi:nil="true"/>
      </GetPolicies>
   </s:Body>
</s:Envelope>
```

Response

After the user is authenticated, the web service retrieves the certificate template that the user should enroll with and creates enrollment policies based on the certificate template properties. A sample of the response can be found on MSDN.

MS-XCEP supports very flexible enrollment policies using various Complex Types and Attributes. For Windows Phone, we will first support the minimalKeyLength, the hashAlgorithmOIDReference policies, and the CryptoProviders. The hashAlgorithmOIDReference has related OID and OIDReferenceID and policySchema in the GetPolicesResponse. The policySchema refers to the certificate template version. Version 3 of MS-XCEP supports hashing algorithms.

Note that the HTTP server response must not be chunked; it must be sent as one message.

Header:

```
HTTP/1.1 200 OK
Date: Fri, 03 Aug 2012 20:00:00 GMT
Server: <sever name here>
Content-Type: application/soap+xml
Content-Length: xxxx
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<s:Envelope
   xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
   xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action s:mustUnderstand="1">

http://schemas.microsoft.com/windows/pki/2009/01/enrollmentpolicy/IPolicy/GetPoliciesResponse
      </a:Action>
      <ActivityId CorrelationId="08d2997e-e8ac-4c97-a4ce-d263e62186ab"
         xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
         d4335d7c-e192-402d-b0e7-f5d550467e3c</ActivityId>
      <a:RelatesTo>urn:uuid: 69960163-adad-4a72-82d2-bb0e5cff5598</a:RelatesTo>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <GetPoliciesResponse
         xmlns="http://schemas.microsoft.com/windows/pki/2009/01/enrollmentpolicy">
         <response>
            <policyFriendlyName xsi:nil="true"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
            <nextUpdateHours xsi:nil="true"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
            <policiesNotChanged xsi:nil="true"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
            <policies>
               <policy>
                  <policyOIDReference>0</policyOIDReference>
```

```
                <cAs xsi:nil="true" />
                <attributes>
                   <policySchema>3</policySchema>
                   <privateKeyAttributes>
                      <minimalKeyLength>2048</minimalKeyLength>
                      <keySpec xsi:nil="true" />
                      <keyUsageProperty xsi:nil="true" />
                      <permissions xsi:nil="true" />
                      <algorithmOIDReference xsi:nil="true" />
                      <cryptoProviders xsi:nil="true" />
                   </privateKeyAttributes>
                   <supersededPolicies xsi:nil="true" />
                   <privateKeyFlags xsi:nil="true" />
                   <subjectNameFlags xsi:nil="true" />
                   <enrollmentFlags xsi:nil="true" />
                   <generalFlags xsi:nil="true" />
                   <hashAlgorithmOIDReference>0</hashAlgorithmOIDReference>
                   <rARequirements xsi:nil="true" />
                   <keyArchivalAttributes xsi:nil="true" />
                   <extensions xsi:nil="true" />
                </attributes>
             </policy>
          </policies>
       </response>
       <cAs xsi:nil="true" />
       <oIDs>
          <oID>
             <value>1.3.14.3.2.29</value>
             <group>1</group>
             <oIDReferenceID>0</oIDReferenceID>
             <defaultName>szOID_OIWSEC_sha1RSASign</defaultName>
          </oID>
       </oIDs>
    </GetPoliciesResponse>
 </s:Body>
</s:Envelope>
```

SOAP faults

If the web service cannot process the request, a SOAP fault is returned with specific fault code and reason.

Fault code	Reason
MessageFormatFault	GetPolicies format is invalid
AuthenticationFault	Failed authentication
AuthorizationFault	Failed authorization
InternalServerFault	Internal error such as SQL down
ClientVersionFault	Unsupported version of client

Certificate enrollment web service

Description

This web service implements the MS-WSTEP protocol. It processes the RequestSecurityToken (RST) message from the client, authenticates the client, requests the certificate from the CA, and returns it in the

RequestSecurityTokenResponse (RSTR) to the client. Besides the issued certificate, the response also contains configurations needed to provision the DM client.

Request

The RequestSecurityToken (RST) must have the user credential and a certificate request. The user credential in an RST soap envelop is the same as in GetPolicies and could be different depends on whether authentication policy is OnPremise or Federated. The BinarySecurityToken in an RST soap body contains a Base64-encoded PKCS#10 certificate request, which is generated by the client based on the enrollment policy. The client could have requested an enrollment policy by using MS-XCEP before requesting a certificate using MS-WSTEP. If the PKCS#10 certificate request is accepted by the certification authority (CA) (the key length, hashing algorithm, etc. match the certificate template), the client can enroll successfully.

Note that the RequestSecurityToken will use a custom TokenType (http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken), because our enrollment token is more than an X.509 v3 certificate. For more details, see the Response section.

The RST may also specify a number of AdditionalContext items, such as DeviceType and Version. Based on these values, for example, the web service can return phone-specific and version-specific DM configuration.

Note that the policy service and the enrollment service must be on the same server; that is, they must have the same host name.

NOTE: In Windows Phone 8 and Windows Phone 8.1's enrollment client PKCS#10 cert request, the CN value has a zero terminator, e.g. B1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00. The server must replace this value in the supplied client certificate. If your server returns a client certificate containing the same Subject value, this can cause unexpected behavior. The server should always override the subject value and not use the default device-provided Device ID Subject= Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00.

Here is a sample RST message (for OnPremise auth policy) to illustrate the details.

Header:

```
POST /EnrollmentServer/DeviceEnrollmentWebService.svc HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
User-Agent: Windows Phone 8 Enrollment Client
Host: enrolltest.contoso.com
Content-Length: 3242
Cache-Control: no-cache
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing"
   xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"
   xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
   xmlns:ac="http://schemas.xmlsoap.org/ws/2006/12/authorization">
   <s:Header>
      <a:Action s:mustUnderstand="1">
         http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RST/wstep
      </a:Action>
      <a:MessageID>urn:uuid:0d5a1441-5891-453b-becf-a2e5f6ea3749</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
```

```
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://enrolltest.contoso.com:443/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
      </a:To>
      <wsse:Security s:mustUnderstand="1">
         <wsse:UsernameToken u:Id="uuid-cc1ccc1f-2fba-4bcf-b063-ffc0cac77917-4">
            <wsse:Username>user@contoso.com</wsse:Username>
            <wsse:Password wsse:Type=
              "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">mypassword
            </wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </s:Header>
   <s:Body>
      <wst:RequestSecurityToken>
         <wst:TokenType>
    http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
         </wst:TokenType>
         <wst:RequestType>
            http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
         <wsse:BinarySecurityToken
            ValueType="http://schemas.microsoft.com/windows/pki/2009/01/enrollment#PKCS10"
            EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd#base64binary">
            DER format PKCS#10 certificate request in Base64 encoding Insterted Here
         </wsse:BinarySecurityToken>
         <ac:AdditionalContext xmlns="http://schemas.xmlsoap.org/ws/2006/12/authorization">
            <ac:ContextItem Name="DeviceType">
               <ac:Value>WindowsPhone</ac:Value>
            </ac:ContextItem>
            <ac:ContextItem Name="ApplicationVersion">
               <ac:Value>8.0.9846.0</ac:Value>
            </ac:ContextItem>
         </ac:AdditionalContext>
      </wst:RequestSecurityToken>
   </s:Body>
</s:Envelope>
```

Here is a sample RST message (for Federated auth policy) to illustrate the details.

Header:

```
POST /EnrollmentServer/DeviceEnrollmentWebService.svc HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
User-Agent: Windows Phone 8 Enrollment Client
Host: enrolltest.contoso.com
Content-Length: 3242
Cache-Control: no-cache
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing"
   xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"
   xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
   xmlns:ac="http://schemas.xmlsoap.org/ws/2006/12/authorization">
   <s:Header>
      <a:Action s:mustUnderstand="1">
         http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RST/wstep
```

```
      </a:Action>
      <a:MessageID>urn:uuid:0d5a1441-5891-453b-becf-a2e5f6ea3749</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://enrolltest.contoso.com:443/ENROLLMENTSERVER/DEVICEENROLLMENTWEBSERVICE.SVC
      </a:To>
      <wsse:Security s:mustUnderstand="1">
         <wsse:BinarySecurityToken  wsse:ValueType=
http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentUserToken
                  wsse:EncodingType=
   http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd#base64binary"
                  xmlns=
          "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                  B64EncodedSampleBinarySecurityToken
               </wsse:BinarySecurityToken>
      </wsse:Security>
   </s:Header>
   <s:Body>
      <wst:RequestSecurityToken>
         <wst:TokenType>
    http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
         </wst:TokenType>
         <wst:RequestType>
            http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
         <wsse:BinarySecurityToken
            ValueType="http://schemas.microsoft.com/windows/pki/2009/01/enrollment#PKCS10"
            EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd#base64binary">
            DER format PKCS#10 certificate request in Base64 encoding Insterted Here
         </wsse:BinarySecurityToken>
         <ac:AdditionalContext xmlns="http://schemas.xmlsoap.org/ws/2006/12/authorization">
            <ac:ContextItem Name="DeviceType">
               <ac:Value>WindowsPhone</ac:Value>
            </ac:ContextItem>
            <ac:ContextItem Name="ApplicationVersion">
               <ac:Value>8.0.9846.0</ac:Value>
            </ac:ContextItem>
         </ac:AdditionalContext>
      </wst:RequestSecurityToken>
   </s:Body>
</s:Envelope>
```

Response

After validating the request, the web service looks up the assigned certificate template for the client, update it if needed, sends the PKCS#10 requests to the CA, processes the response from the CA, constructs an OMA Client Provisioning XML format, and returns it in the RequestSecurityTokenResponse (RSTR).

Note 1: The HTTP server response must not be chunked; it must be sent as one message.

Note 2: Do NOT use Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00. The server must replace this value in the supplied client certificate. If your server returns a client certificate containing the same Subject value, this can cause unexpected behavior. The server should always override the subject value and not use the default device-provided Device ID Subject= Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00

in the supplied client certificate and certificate search criteria provisioning nodes (SSLCLIENTCERTSEARCHCRITERIA in APPLICATION Configuration Service Provider, and in CertificateSearchCriteria node EnterpriseAppManagement configuration service provider).

Similar to the TokenType in the RST, the RSTR will use a custom ValueType in the BinarySecurityToken (http://schemas.microsoft.com/ConfigurationManager/Enrollment/DeviceEnrollmentProvisionDoc), because the token is more than an X.509 v3 certificate.

The provisioning XML contains:

- (mandatory) The requested certificates
- (mandatory) The DM client configuration
- (optional) An enterprise application token and an enterprise app download link to allow the enrollment client to download a Company Hub or enterprise app at the end of enrollment

The client will install the client certificate, the enterprise root certificate, and intermediate CA certificate if there is one. The DM configuration includes the name and address of the DM server, which client certificate to use, and schedules when the DM client calls back to the server.

NOTE 1: Enrollment provisioning XML should contain a maximum of one root certificate and one intermediate CA certificate that is needed to chain up the MDM client certificate. Additional root and intermediate CA certificates could be provisioned during an OMA DM session.

NOTE 2: When provisioning root and intermediate CA certificates, the supported CSP node path is: CertificateStore/Root/System for root certificate provisioning, CertificateStore/CA/System for intermediate CA certificate provisioning.

Here is a sample RSTR message and a sample of OMA client provisioning XML within RSTR. For more information about the configuration service providers (CSPs) used in provisioning XML, see the Enterprise settings, policies and app management section.

RSTR message:

Header:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 10231
Content-Type: application/soap+xml; charset=utf-8
Server: Microsoft-IIS/7.0
Date: Fri, 03 Aug 2012 00:32:59 GMT

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:a="http://www.w3.org/2005/08/addressing"
   xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <s:Header>
      <Action s:mustUnderstand="1" >
         http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RSTRC/wstep
      </Action>
      <a:RelatesTo>urn:uuid:81a5419a-496b-474f-a627-5cdd33eed8ab</a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o=
         "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2012-08-02T00:32:59.420Z</u:Created>
            <u:Expires>2012-08-02T00:37:59.420Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
```

```
      <RequestSecurityTokenResponseCollection
         xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
         <RequestSecurityTokenResponse>
            <TokenType>
    http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
            </TokenType>
            <RequestedSecurityToken>
               <BinarySecurityToken
                  ValueType=
"http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentProvisio
nDoc"
                  EncodingType=
   "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd#base64binary"
                  xmlns=
          "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                  B64EncodedSampleBinarySecurityToken
               </BinarySecurityToken>
            </RequestedSecurityToken>
            <RequestID xmlns="http://schemas.microsoft.com/windows/pki/2009/01/enrollment">0
            </RequestID>
         </RequestSecurityTokenResponse>
      </RequestSecurityTokenResponseCollection>
   </s:Body>
</s:Envelope>
```

Sample provisioning XML (presented in the preceding package as a security token):

```
<wap-provisioningdoc version="1.1">
   <characteristic type="CertificateStore">
      <characteristic type="Root">
         <characteristic type="System">
            <characteristic type="031336C933CC7E228B88880D78824FB2909A0A2F">
               <parm name="EncodedCertificate" value="B64 encoded cert insert here" />
            </characteristic>
         </characteristic>
      </characteristic>
      <characteristic type="My" >
         <characteristic type="User">
            <characteristic type="F9A4F20FC50D990FDD0E3DB9AFCBF401818D5462">
               <parm name="EncodedCertificate" value="B64EncodedCertInsertedHere" />
            </characteristic>
            <characteristic type="PrivateKeyContainer"/>
            <!-- This tag must be present for XML syntax correctness. -->
         </characteristic>
         <characteristic type="WSTEP">
            <characteristic type="Renew">
             <!—If the datatype for ROBOSupport, RenewPeriod, and RetryInterval tags exist,
they must be set explicitly. -->
               <parm name="ROBOSupport" value="true" datatype="boolean"/>
               <parm name="RenewPeriod" value="60" datatype="integer"/>
               <parm name="RetryInterval" value="4" datatype="integer"/>
            </characteristic>
         </characteristic>
      </characteristic>
   </characteristic>
   <characteristic type="APPLICATION">
      <parm name="APPID" value="w7"/>
      <parm name="PROVIDER-ID" value="TestMDMServer"/>
      <parm name="NAME" value="Microsoft"/>
      <parm name="ADDR" value="https://DM.contoso.com:443/omadm/WindowsPhone.ashx"/>
```

```
      <parm name="CONNRETRYFREQ" value="6" />
      <parm name="INITIALBACKOFFTIME" value="30000" />
      <parm name="MAXBACKOFFTIME" value="120000" />
      <parm name="BACKCOMPATRETRYDISABLED" />
      <parm name="DEFAULTENCODING" value="application/vnd.syncml.dm+wbxml" />
      <parm name="SSLCLIENTCERTSEARCHCRITERIA" value=
  "Subject=DC%3dcom%2cDC%3dmicrosoft%2cCN%3dUsers%2cCN%3dAdministrator&amp;Stores=My%5CUser"/>
      <characteristic type="APPAUTH">
         <parm name="AAUTHLEVEL" value="CLIENT"/>
         <parm name="AAUTHTYPE" value="DIGEST"/>
         <parm name="AAUTHSECRET" value="password1"/>
         <parm name="AAUTHDATA" value="B64encodedBinaryNonceInsertedHere"/>
      </characteristic>
      <characteristic type="APPAUTH">
         <parm name="AAUTHLEVEL" value="APPSRV"/>
         <parm name="AAUTHTYPE" value="BASIC"/>
         <parm name="AAUTHNAME" value="testclient"/>
         <parm name="AAUTHSECRET" value="password2"/>
      </characteristic>
   </characteristic>
   <characteristic type="DMClient"> <!-- Staring with Windows Phone 8.1, an enrollment server
should use DMClient CSP XML to configure DM polling schedules. The polling schedule regisitry
keys will be deprecated after Windows Phone 8.1.-->
      <characteristic type="Provider">
<!-- ProviderID in DMClient CSP must match to PROVIDER-ID in w7 APPLICATION characteristics --
>
       <characteristic type="TestMDMServer">
 <characteristic type="Poll">
         <parm name="NumberOfFirstRetries" value="8" datatype="integer" />
         <parm name="IntervalForFirstSetOfRetries" value="15" datatype="integer" />
         <parm name="NumberOfSecondRetries" value="5" datatype="integer" />
         <parm name="IntervalForSecondSetOfRetries" value="3" datatype="integer" />
         <parm name="NumberOfRemainingScheduledRetries" value="0" datatype="integer" />
<!-- In Windows Phone 8.1, MDM push is supported for real-time communication. The DM client
long term polling schedule's retry waiting interval should be more than 24 hours (1440) to
reduce the impact to data consumption and battery life. Refer to the DMClient Configuration
Service Provider section for information about polling schedule parameters.-->
         <parm name="IntervalForRemainingScheduledRetries" value="1560" datatype="integer" />
 </characteristic>
       <parm name="EntDeviceName" value="Administrator_WindowsPhone" datatype="string" />
</characteristic>
      </characteristic>
   </characteristic>
   <characteristic type="EnterpriseAppManagement">
      <characteristic type="1">
         <parm datatype="string" name="EnrollmentToken" value="AppEnrollTokenInsertedHere"/>
         <parm datatype="string" name="StoreProductId"
            value="{92A7F577-6F01-243F-8399-088E0DC40656}"/>
         <parm datatype="string" name="StoreURI"

value="HTTPS://DM.contoso.com:443/EnrollmentServer/clientcabs/EnterpriseApp1.xap"/>
         <parm datatype="string" name="StoreName" value="Contoso App Store"/>
<!-- The value must be a URL encoded representation of the X.500 destinguished name of the
client certificates Subject property. -->
         <parm datatype="string" name="CertificateSearchCriteria" value="
SearchCriteriaInsertedHere"/>
         <parm datatype="string" name="CRLCheck" value="0"/>
      </characteristic>
   </characteristic>
</wap-provisioningdoc>
```

NOTE 1: parm name and characteristic type in w7 APPLICATION CSP XML are case sensitive and must be all uppercase.

NOTE 2: In w7 APPLICATION characteristic, both CLIENT and APPSRV credentials should be provided in XML.

For detailed descriptions of these settings, see the Enterprise settings, policies and app management section, later in this document.

Note that the PrivateKeyContainer characteristic is required and must be present in the Enrollment provisioning XML by the enrollment. Other important settings are the PROVIDER-ID, NAME, and ADDR parameter elements, which need to contain the unique ID and NAME of your DM provider and the address where the phone can connect for configuration provisioning. The ID and NAME can be arbitrary values, but they must be unique.

Also important is the **SSLCLIENTCERTSEARCHCRITERIA**, which is used for selecting the certificate to be used for client authentication. The search is based on the subject attribute of the signed user certificate.

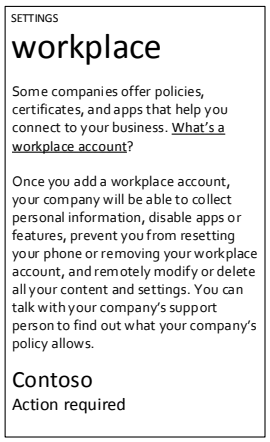
Request for certificate renewal

The enrolled client certificate expires after a period of use. The expiration date is specified by the server. To ensure continuous access to enterprise applications, the phone supports a user-triggered certificate renewal process. The user is prompted to provide the current password for the corporate account, and the enrollment client gets a new client certificate from the enrollment server and deletes the old certificate. The client generates a new private/public key pair, generates a PKCS#7 request, and signs the PKCS#7 request with the existing certificate. In Windows Phone 8.1, automatic MDM client certificate renewal is also supported. Refer following section for more details.

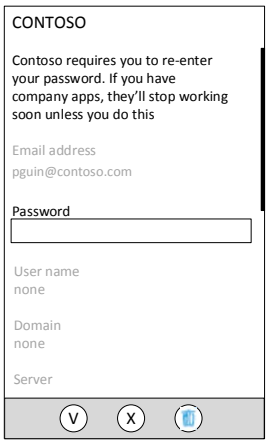
> Note: Make sure that the EntDMID in the DMClient configuration service provider is set before the certificate renewal request is triggered.

Certificate manual renewal UI

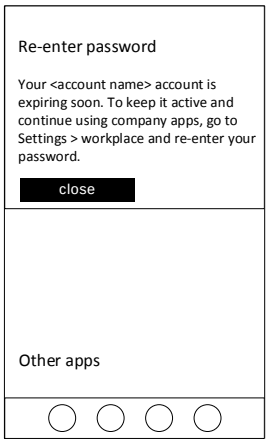
The user experience is the enrollment client's built-in UI, and there is no third-party extensibility. Before the client certificate expires, the phone notifies the user to go to the workplace settings page to renew the account. When the user navigates to the workplace setting page and taps the account name, the account detail screen appears. When the user provides the updated corporate password, the enrollment client communicates with the enrollment server to get the updated certificate.



Tap Attention required



Enter password; tap Done



Cert expiring warning dialog

In detail, the client creates a GetPolicies request and processes the GetPoliciesResponse as before. Then, the client creates an RST renewal request to retrieve a new certificate from the enrollment web service. The renewal request specifies a different RequestType from the initial enrollment request (Renew instead of Issue). It also uses a different BinarySecurityToken ValueType (PKCS#7 instead of PKCS#10).

Here is a sample to illustrate the details of a manual renewal request.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u=
   "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action s:mustUnderstand="1">
         http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RST/wstep</a:Action>
      <a:MessageID>urn:uuid:61a17f2c-42e9-4a45-9c85-f15c1c8baee8</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://dm.contoso.com/EnrollmentService/DeviceEnrollmentService.svc</a:To>
      <o:Security s:mustUnderstand="1" xmlns:o=
         "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2011-07-11T19:49:08.579Z</u:Created>
            <u:Expires>2011-07-11T19:54:08.579Z</u:Expires>
         </u:Timestamp>
         <o:UsernameToken u:Id="uuid-2a734df6-b227-4e60-82a8-ed53c574b718-5">
            <o:Username>user@contoso.com</o:Username>
            <o:Password o:Type=
               "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">
               7Apples
            </o:Password>
         </o:UsernameToken>
      </o:Security>
   </s:Header>
   <s:Body>
      <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
         <TokenType>
    http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
         </TokenType>
```

```
        <RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew</RequestType>
        <BinarySecurityToken
           ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd#PKCS7"
           EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd#base64binary"
           xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">
           BinarySecurityTokenInsertedHere
        </BinarySecurityToken>
        <AdditionalContext xmlns="http://schemas.xmlsoap.org/ws/2006/12/authorization">
           <ContextItem Name="DeviceType">
              <Value>WindowsPhone</Value>
           </ContextItem>
           <ContextItem Name="ApplicationVersion">
              <Value>5.0.7616.0</Value>
           </ContextItem>
        </AdditionalContext>
      </RequestSecurityToken>
   </s:Body>
</s:Envelope>
```

Automatic MDM client certificate renew via Renew On Behalf Of (ROBO) function in WSTEP

In addition to manual certificate renew, Windows Phone 8.1 adds support for automatic certificate renew (ROBO – Renew On Behalf Of) that does not require any user interaction. For auto renew, the enrollment client uses the existing MDM client certificate to perform client Transport Layer Security (TLS). The user security token is not needed in the SOAP header. As a result, the MDM certificate enrollment server is required to support client TLS for certificate based client authentication for automatic certificate renew.

Note 1: Auto certificate renew is the only supported MDM client certficatecertificate renew method for the device that is enrolled via WAB authentication (AuthPolicy = Federated). It also means if the server supports WAB authentication, the MDM certificate enrollment server MUST also support client TLS in order to renew the MDM client certificate.

Note 2: For the device that is enrolled with the OnPremise authentication method, for backward compatibility, the default renew method is user manual certificate renew. However, for Windows Phone 8.1 device, during the MDM client certificate enrollment phase or during MDM management section, the enrollment server or MDM server could configure the device to support automatic MDM client certificate renew via CertificateStore CSP's ROBOSupport node under CertificateStore/My/WSTEP/Renew URL. For information about Renew related configuration settings, refer CertificateStore configuration service provider (Updated in Windows Phone 8.1)

Note 3: Unlike manual certificate renew where there is an additional b64 encoding for PKCS#7 message content, with automatic renew, the PKCS#7 message content isn't b64 encoded separately.

Note 4: During the automatic cert renew process, if the root certificate isn't trusted by the device, the authentication will fail. Make sure using one of device pre-installed root certificates or provision the root cert over a DM session via CertificateStore Configuration Service Provider.

Note 5: During the automatic cert renew process, the device will deny HTTP redirect request from the server unless it is the same redirect URL that the user explicitly accepted during the initial MDM enrollment process.

Note 6: The renewal process follows the same steps as device enrollment, which means that it starts with Discovery service, followed by Enrollment policy service, and then Enrollment web service.

Here is a sample to illustrate the details of an automatic certificate renewal request.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u=
   "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action s:mustUnderstand="1">
         http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RST/wstep</a:Action>
      <a:MessageID>urn:uuid:61a17f2c-42e9-4a45-9c85-f15c1c8baee8</a:MessageID>
      <a:ReplyTo>
         <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To s:mustUnderstand="1">
         https://dm.contoso.com/EnrollmentService/DeviceEnrollmentService.svc</a:To>
      <o:Security s:mustUnderstand="1" xmlns:o=
         "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2011-07-11T19:49:08.579Z</u:Created>
            <u:Expires>2011-07-11T19:54:08.579Z</u:Expires>
         </u:Timestamp>
         <o:UsernameToken u:Id="uuid-2a734df6-b227-4e60-82a8-ed53c574b718-5">
            <o:Username>user@contoso.com</o:Username>
            <o:Password o:Type=
               "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">
            </o:Password>
         </o:UsernameToken>
      </o:Security>
   </s:Header>
   <s:Body>
      <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
         <TokenType>
    http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
         </TokenType>
         <RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew</RequestType>
         <BinarySecurityToken
            ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd#PKCS7"
            EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd#base64binary"
            xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">
            BinarySecurityTokenInsertedHere
         </BinarySecurityToken>
         <AdditionalContext xmlns="http://schemas.xmlsoap.org/ws/2006/12/authorization">
            <ContextItem Name="DeviceType">
               <Value>WindowsPhone</Value>
            </ContextItem>
            <ContextItem Name="ApplicationVersion">
               <Value>5.0.7616.0</Value>
            </ContextItem>
         </AdditionalContext>
      </RequestSecurityToken>
   </s:Body>
</s:Envelope>
```

Certificate renew schedule configuration

In Windows Phone 8, the renew period can only be set during the MDM enrollment phase. Windows Phone 8.1 supports a certificate renew period and renew failure retry to be configurable by both MDM enrollment server and later by the MDM management server via CertificateStore CSP's RenewPeriod and RenewInterval nodes. The device could retry automatic certificate renew multiple time till the certificate is expired. For manual certificate renew, instead of only reminding the user once as in Windows Phone 8, the Windows Phone 8.1 device will remind the user with a prompt dialog at every renew retry time until the certificate is expired.

The renewal process follows the same steps as device enrollment, which means that it starts with Discovery service, followed by Enrollment policy service, and then Enrollment web service.

For more information about the parameters, see CertificateStore configuration service provider (Updated in Windows Phone 8.1).

Note: Unlike manually cert renew, the device will not perform an automatic MDM client cert renewal if the cert is already expired. To make sure that the device has enough time to perform an automatic renewal, we recommend that you set a renewal period a couple months (40-60 days) before the certificate expires and set the renewal retry interval to be every few days such as every 4-5 days instead every 7 days (weekly) to increase the chance that the device will a connectivity at different days of the week.

Updateability consideration

When the user updates their MDM enrolled Windows Phone 8 device to Windows Phone 8.1, for backward compatibility, devices enrolled with OnPremise authentication will continue to use the user manual certificate renew method (unless the MDM server configure the updated the device to support automatic cert renew later). The only difference is instead of prompting the user only one time for account updating, the device will use default renew retry interval (once a week) to remind the user multiple times till cert is renewed.

If the certificate is already expired when the user updates the Windows Phone 8 device to Windows Phone 8.1, unlike in Windows Phone 8, there is no more warning dialog. However the user may still go to "company apps" to provide an updated password and try renew again. Windows Phone 8.1 will prompt a dialog warning the user the certificate is expiring and user should provide updated password to try to renew again. The enrollment server can make a decision on whether accepting a manual renew request.

For Windows Intune managed devices, if the certtificate is already expired when the user updates the Windows Phone 8 device to Windows Phone 8.1, automatic cert renew would kick in and send renew request to the server on behalf of the user. If the server accepts the request, the certificate will be installed. Note this is only for updating case and is different from normal Windows Phone 8.1 automatic certificate renew which stops sending renew request if certificate is expired.

Response for certificate renewal

When RequestType is Renew, the web service verifies the following (in additional to initial enrollment):

- The signature of the PKCS#7 BinarySecurityToken is correct
- The client's certificate is in the renewal period
- The certificate was issued by the enrollment service
- The requester is the same as the requester for initial enrollment
- For standard client's request, the client hasn't been blocked

After validations, the web service retrieves the PKCS#10 content from the PKCS#7 BinarySecurityToken. The rest is the same as initial enrollment, except that the Provisioning XML only needs to have the new certificate issued by the CA.

Note that the HTTP server response must not be chunked; it must be sent as one message.

```
<wap-provisioningdoc version="1.1">
   <characteristic type="CertificateStore">
<!-- Root certificate provision is only needed here if it is not in the device already -->
<characteristic type="Root">
         <characteristic type="System">
            <characteristic type="EncodedRootCertHashInsertedHere ">
               <parm name="EncodedCertificate" value="EncodedCertInsertedHere" />
            </characteristic>
         </characteristic>
      </characteristic>
      <characteristic type="My" >
         <characteristic type="User">
            <characteristic type="EncodedClientCertHashInsertedHere">
               <parm name="EncodedCertificate" value="EncodedCertInsertedHere" />
               <characteristic type="PrivateKeyContainer"/>
            </characteristic>
         </characteristic>
      </characteristic>
   </characteristic>
   <characteristic type="APPLICATION">
      <parm name="PROVIDER-ID" value="TestMDMServer"/>
   </characteristic>
</wap-provisioningdoc>
```

Note that the client receives a new certificate, instead of renewing the initial certificate. The administrator controls which certificate template the client should use. The templates may be different at renewal time than the initial enrollment time. Issuing a new certificate using the template at renewal time honors the administrator's latest intention.

Configuration service providers supported during MDM enrollment and certificate renewal

The following configuration service providers are supported during MDM enrollment and certificate renewal process. See Configuration service provider reference for detailed descriptions of each configuration service provider.

- CertificateStore configuration service provider
- w7 APPLICATION configuration service provider
- DMClient configuration service provider
- EnterpriseAppManagement configuration service provider

SOAP faults

If the web service cannot process the request, a SOAP fault is returned with specific fault code and reason.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u=
   "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action s:mustUnderstand="1" xmlns:a="http://www.w3.org/2005/08/addressing"
         xmlns:s="http://www.w3.org/2003/05/soap-envelope">
         http://www.w3.org/2005/08/addressing/soap/fault
      </a:Action>
```

```
      <a:RelatesTo xmlns:a="http://www.w3.org/2005/08/addressing">
         urn:uuid:2d37bdb7-e4ac-4bb8-bca3-29cc9f5cf6b4
      </a:RelatesTo>
      <o:Security s:mustUnderstand="1" xmlns:o=
         "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2012-9-27T04:20:03.408Z</u:Created>
            <u:Expires>2012-9-27T04:25:03.408Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body>
      <s:Fault>
         <s:Code>
            <s:Value>s:Receiver</s:Value>
            <s:Subcode>
               <s:Value>s:MessageFormat</s:Value>
            </s:Subcode>
         </s:Code>
         <s:Reason>
            <s:Text xml:lang="en-US">Invalid TokenType in request</s:Text>
         </s:Reason>
      </s:Fault>
   </s:Body>
</s:Envelope>
```

Fault code	Reason
MessageFormatFault	RST format is invalid
AuthenticationFault	Failed authentication
AuthorizationFault	Failed authorization
CertificateRequestFault	CA denied cert request
InternalServerFault	Internal error such as SQL down
InvalidRenewalRequesterFault	Request for renewal is different from initial enrollment requester
RenewalWindowFault	Cert not in renewal window
ClientVersionFault	Unsupported version of client
NotReachedRenewalWindow	Renewal request isn't within the renewal window. Requester: user thumbprint: <*thumbprint inserted here*>

Best practice tips

General notes

All POSTs should have HTTP Content-Type "`application/soap+xml; charset=utf-8`" for the discovery and enrollment phase. For SyncML, the proper content types are either "`application/vnd.syncml.dm+xml`" or "`application/vnd.syncml.dm+wbxml`", depending on the choice made in the provisioning XML.

HTTP 1.1 "Content-Encoding" using "Chunked" is not supported. The server should explicitly specify the "Content-Length" header accordingly in all responses, which turns off the chunked response generation in most frameworks.

Certificates

Web server SSL certificate

The SSL wildcard server certificate is supported for enrollment and MDM session.

The web server certificate must also have certain X.509 v3 extensions before the phone accepts it. The discovery and enrollment phases could have more relaxed criteria for the certificate, but the SyncML Post must have an exact set of extensions present. The following table shows the required extensions.

Extension	Description	Value
Key Usage	Lists the permitted uses of this particular certificate.	Digital Signature, Key Encipherment
Extended Key Usage	Extensions for the key usage.	TLS Web Server Authentication
Subject Alternative Name	Alternative subjects for this certificate. Used for listing the alternate domains for which the certificate can be used.	DNS entries for each subdomain you're using this certificate for. For example, "DNS:secure.mydmpoc.net, DNS:enterpriseenrollment.mydiscovery.net"

To avoid the need to manually install server certificates, it is best if the certificates for the web server, MDM server, and client chain to the same root CA as those that are installed during enrollment or to another root CA already trusted by the phone.

Signed client certificate

The client certificate that is provisioned to the client in the provisioning XML should also have certain X.509 v3 extension. The following extensions are required.

Extension	Description	Value
Key Usage	Lists the permitted uses of this particular certificate.	Digital Signature
Extended Key Usage	Extensions for the key usage.	TLS Web Client Authentication
Subject Key Identifier	Provides a means for identifying certificates that contain a particular public key.	

Note that you should also specify the subject of the certificate in such a way that you can reference the certificate in the provisioning XML's **SSLCLIENTCERTSEARCHCRITERIA** parameter.

Note that when creating a certificate, the server should set meaningful/distinguished common name instead of some well-known GUIDs.

For more information about how to embed the certificate to the provisioning XML, see the Response section of the Certificate enrollment web service section, earlier in this document.

Disconnecting from the management infrastructure (unenrollment)

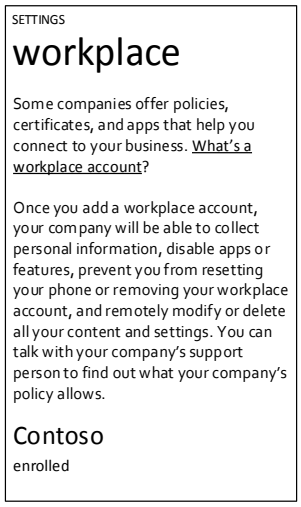
Disconnecting may be initiated either locally by the user from the phone or remotely by the IT admin via management server. User-initiated disconnection is performed much like the initial connection, and it is initiated from the same location in the Setting Control Panel as creating the workplace account. Users may choose to disconnect for any number of reasons, including leaving the company or getting a new phone and no longer needing access to their LOB apps on the old phone. When an admin initiates a disconnection, the enrollment client performs the disconnection during its next regular maintenance session. Admins may choose to disconnect a user's phone after they've left the company or because the phone is regularly failing to comply with the organization's security settings policy.

During disconnection, the client does the following:

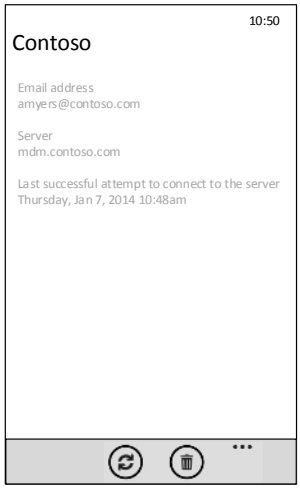
- Removes the enterprise application token that allowed installing and running LOB apps. Any business applications associated with this enterprise token are removed as well.
- Removes certificates that are configured by MDM server.
- Ceases enforcement of the settings policies that the management infrastructure has applied.
- Removes the device management client configuration and other setting configuration added by MDM server, including the scheduled maintenance task. The client remains dormant unless the user reconnects it to the management infrastructure.
- Reports successful initiated disassociation to the management infrastructure if the admin initiated the process. Note that in Windows Phone 8.1, user-initiated disassociation is reported to the server as a best effort.

User-initiated disconnection

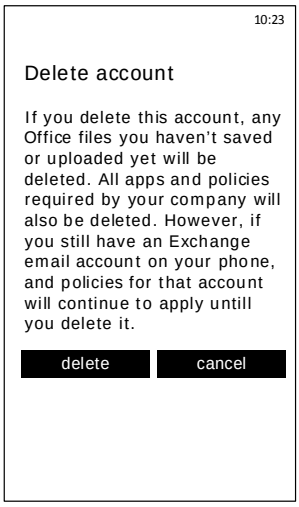
The following mockup shows the user experience of disconnection from enterprise management.



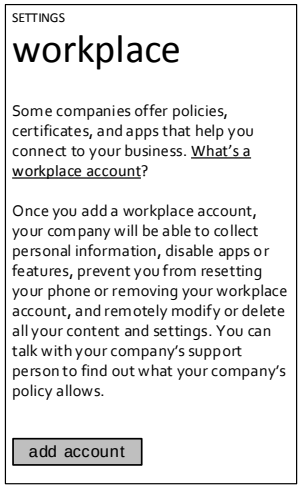
Tap company name **Contoso**



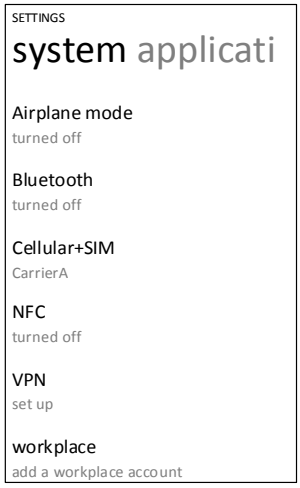
Tap **Trash** icon



Tap **delete** button to confirm



workplace settings screen
(Contoso removed)



System settings screen
(Contoso removed)

User unenrollment notification to the MDM server

In Windows Phone 8 the device will not notify the server when user deletes the organization account. In WP Windows Phone 8.1, after the user confirms the account deletion command and before the account is deleted, the MDM client will send a notification to the MDM server notifying that the server the account will be removed. This is a best effort action as no retry is built-in to ensure the notification is successfully sent to the device.

This leverages the OMA DM generic alert 1226 function to send a user an MDM unenrollment user alert to the MDM server after the device accepts the user un-enroll request, but before it deletes any enterprise data. The server should set expectation that un-enroll may succeed or fail and the server can check whether the device is unenrolled by either checking whether the device calls back at scheduled time or by sending a push notification to the device to see whether it responds back. If the server plans to send push notification, it should allow for some delay to give the device the time to finish the un-enrollment work.

1. User unenroll generic alert definition: 1226 generic Alert is an OMA DM standard. For more information, check OMA Device Management Protocol specification (OMA-TS-DM_Protocol-V1_2_1-20080617-A) , available from the OMA website. Vendor uses Type attribute to specify what type of generic alert it is. For device initiated MDM unenroll, the following alert type is used: com.microsoft:mdm.unenrollment.userrequest

2. Example Flow
    1. User elects to un-enroll
    2. Any active MDM OMA DM sessions are terminated.

3. DM client kicks off DM session, including a user unenroll generic alert in the first package it sends to the server.

Sample OMA DM pkg 1 that contains generic alert message listed below. For more information on WP OMA DM support, check section DM SyncML functionality support

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>1</MsgID>
      <Target>
         <LocURI>{unique device ID}</LocURI>
      </Target>
      <Source>
         <LocURI>https://www.thephone-company.com/mgmt-server</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Alert>
        <CmdID>2</CmdID>
        <Data>1226</Data> <!-- generic alert -->
        <Item>
          <Meta>
             <Type xmlns=”syncml:metinfo”>
com.microsoft:mdm.unenrollment.userrequest</Type>
              <Format xmlns= “syncml:metinfo”>int</Format>
           </Meta>
           <Data>1</Data>
        </Item>
       </Alert>

    <!-- other device information -->
      <Replace>
         <CmdID>2</CmdID>
         <Item>
            <Source>
               <LocURI>./DevInfo/DevID</LocURI>
            </Source>
            <Data>{unique device ID}</Data>
         </Item>
         <Item>
           ...
         </Item>
      </Replace>
      <Final/>
   </SyncBody>
</SyncML>
```

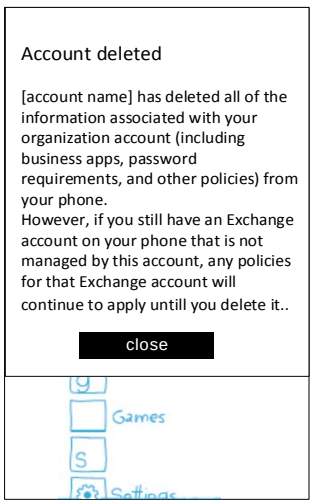
4. Starts the unenrollment process.

IT admin–requested disconnection

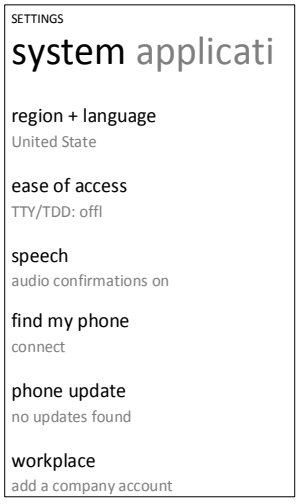
The server requests an enterprise management disconnection request by issuing an Exec OMA DM SyncML XML command to the phone via the DMClient configuration service provider's Unenroll node

during the next client-initiated DM session. The Data tag inside the Exec command should be the value of the provisioned DM server ProviderID. For more details, see Enterprise-specific DM client configuration.

When the disconnection is completed, the user is notified that the phone has been disconnected from enterprise management.



Tap **close** to acknowledge (Contoso removed)



system settings screen (Contoso removed)

workplace settings screen

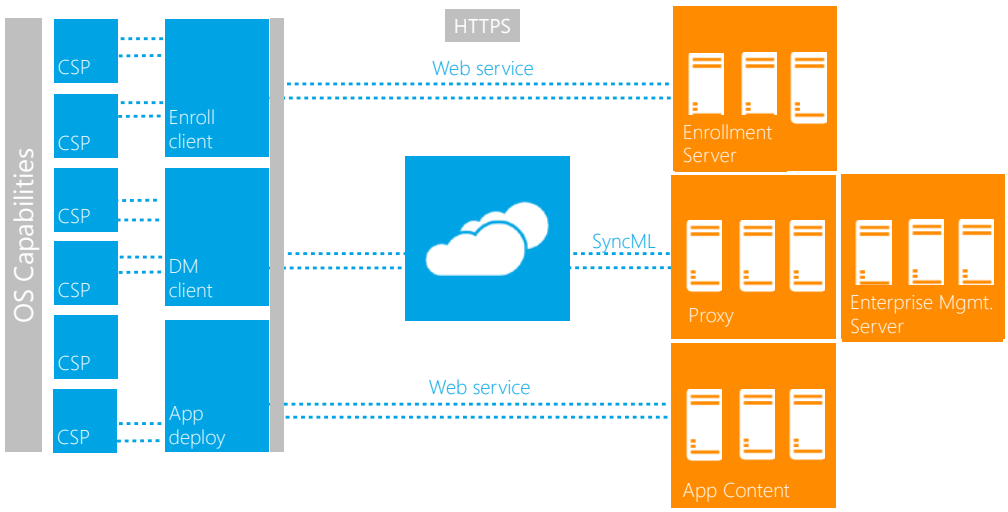
Enterprise settings, policies and app management

The actual management interaction between the phone and server is done via the DM client. The DM client communicates with the enterprise management server via DM v1.2 SyncML syntax. The OMA website provides the protocol details.

Windows Phone 8.1 currently supports one MDM server. The DM client that is configured via the enrollment process is granted access to enterprise related settings. Enterprise MDM settings are exposed via various configuration service providers to the DM client and described in this section.

The DM client is also configured during the enrollment process to be invoked by the task scheduler to periodically poll the MDM server. Additionally, the end user can use Sync the button in Windows Phone settings, under the 'company app' detail page to force the DM client to connect to the server immediately.

The following diagram shows the work flow between server and client.



Management Work Flow

This protocol defines an HTTPS-based client/server communication with DM SyncML XML as the package payload that carries management requests and execution results. The configuration request is addressed via a managed object (MO). The settings supported by the managed object are represented in a conceptual tree structure. This logical view of configurable phone settings simplifies the way the server addresses the phone settings by isolating the implementation details from the conceptual tree structure.

To facilitate security-enhanced communication with the remote server for enterprise management, Windows Phone 8 supports certificate-based mutual authentication over an encrypted SSL HTTP channel between the DM client and management service. The server and client certificates are provisioned during the enrollment process.

DM client configuration, company policy enforcement, business application management, and phone inventory are all exposed or expressed via configuration service providers (CSPs). CSPs are the Windows Phone term for managed objects. The DM client communicates with the server and sends configuration request to CSPs. The server only needs to know the logical local URIs defined by those CSP nodes in order to use the DM protocol XML to manage the phone.

The Windows Phone has similar level of OMA DM protocol v1.2 support as Windows Mobile 6.5. This document focuses on what Windows Phone managed objects are enabled to be accessible by an enterprise DM server.

As a summary, the following are the DM tasks that an organization's management server could support:

- Company policy management: Company policies are supported via the DeviceLock CSP, StorageCard CSP, and Registry CSP for phone encryption status. It enables the management service to configure device lock related policies, disable/enable the storage card, and query phone encryption status. The RemoteWipe CSP allows IT pros to remotely fully wipe the internal user data storage. The DeviceLock and StorageCard CSPs are only accessible by the enterprise service.
- Enterprise application management: This is addressed via the EnterpriseAppManagement CSP. It is used to install the enterprise token, query installed business application names and version, etc. This CSP is only accessible by the enterprise service.

- Certificate management: This is supported via the CertificateStore CSP to install new ROOT and CA certificates
- Basic phone inventory and asset management: Some basic phone information can be retrieved via the DevInfo and DevDetail CSPs. These provide basic phone information such as OEM name, phone model, hardware version, OS version, processor types, etc. This is for asset management and phone targeting**.** The NodeCache CSP enables the phone to only send out delta inventory settings to the server to reduce over-the-air data usage. The NodeCache CSP is only accessible by the enterprise service.

DM SyncML functionality support

This section describes the OMA DM functionalities that the Windows Phone DM client supports in general. Note that for enterprise device management not all OMA DM client functions are needed. For example, server notification over WAP Push via binary SMS is used by the mobile operator but isn't used by the enterprise server. The full description of the OMA DM protocol v1.2 can be found at the OMA website.

OMA DM standards

The following table shows the OMA DM standards that Windows Phone uses.

General area	OMA DM standard that is supported
Data transport and session	• Client-initiated remote HTTPS DM session over SSL.
Bootstrap XML	• OMA Client Provisioning profile.

General area	OMA DM standard that is supported
DM protocol commands	The following list shows the commands that are used by the phone. For further information about the OMA DM command elements, see "SyncML Representation Protocol Device Management Usage (OMA-SyncML-DMRepPro-V1_1_2-20030613-A)" available from the OMA website. • Add (Implicit Add supported) • Alert (DM alert): server-initiated management alert (1200) (not used by enterprise management), session abort (1223), UI Alerts (1100, 1101, 1102, 1103, 1104) (not used by enterprise management), generic alert (1226) (only used by enterprise management client when the user triggers a MDM unenrollment action from the device) • Atomic: Note that performing an Add command followed by Replace on the same node within an Atomic element is not supported. Nested Atomic and Get commands are not allowed and will generate error code 500. • Delete: Removes a node from the DM tree, and the entire subtree beneath that node if one exists • Exec: Invokes an executable on the phone • Get: Retrieves data from the phone; for interior nodes, the child node names in the Data element are returned in URI-encoded format • Replace: Overwrites data on the phone • Result: Returns the data results of a Get command to the DM server • Sequence: Specifies the order in which a group of commands must be processed • Status: Indicates the completion status (success or failure) of an operation If an XML element that is not a valid OMA DM command is under one of the following elements, the status code 400 is returned for that element: • SyncBody • Atomic • Sequence If no CmdID is provided in the DM command, the client returns blank in the status element and the status code 400. If Atomic elements are nested, the following status codes are returned: • The nested Atomic command returns 500. • The parent Atomic command returns 507. Note: Performing an Add command followed by Replace on the same node within an Atomic element is not supported. LocURI cannot start with "/". Meta XML tag in SyncHdr is ignored by the phone.
OMA DM standard objects	• DevInfo • DevDetail • OMA DM DMS account objects (OMA DM version 1.2)

General area	OMA DM standard that is supported
Security	• Authenticate DM server initiation notification SMS message (not used by enterprise management) • Application layer Basic and MD5 client authentication • Authenticate server with MD5 credential at application level • Data integrity and authentication with HMAC at application level • SSL level certificate-based client/server authentication, encryption, and data integrity check
Nodes	In the OMA DM tree, the following rules apply for the node name: • "." can be part of the node name. • The node name cannot be empty. • The node name cannot be only the asterisk (*) character.
Provisioning files	Provisioning XML must be well formed and follow the definition in SyncML Representation Protocol specification. If an XML element that is not a valid OMA DM command is under SyncBody, the status code 400 is returned for that element. Note: To represent a Unicode string as a URI, first encode the string as UTF-8. Then encode each of the UTF-8 bytes using URI encoding.
WBXML support	Windows Phone supports sending and receiving SyncML in both XML format and encoded WBXML format. This is configurable by using the DEFAULTENCODING node under the w7 APPLICATION characteristic during enrollment. For more information about WBXML encoding, see section 8 of the SyncML Representation Protocol specification.

OMA DM protocol common elements

Common elements are used by other OMA DM element types. The following table lists the OMA DM common elements that are used to configure Windows Phones. For more information about OMA DM common elements, see "SyncML Representation Protocol Device Management Usage" (OMA-SyncML-DMRepPro-V1_1_2-20030613-A) available from the OMA website.

Element	Description
Chal	Specifies an authentication challenge. The server or client can send a challenge to the other if no credentials or inadequate credentials were given in the original request message.
Cmd	Specifies the name of an OMA DM command referenced in a Status element.
CmdID	Specifies the unique identifier for an OMA DM command.
CmdRef	Specifies the ID of the command for which status or results information is being returned. This element takes the value of the CmdID element of the corresponding request message.
Cred	Specifies the authentication credential for the originator of the message.
Final	Indicates that the current message is the last message in the package.
LocName	Specifies the display name in the Target and Source elements, used for sending a user ID for MD5 authentication.

Element	Description
LocURI	Specifies the address of the target or source location.
MsgID	Specifies a unique identifier for an OMA DM session message.
MsgRef	Specifies the ID of the corresponding request message. This element takes the value of the request message MsgID element.
RespURI	Specifies the URI that the recipient must use when sending a response to this message.
SessionID	Specifies the identifier of the OMA DM session associated with the containing message.
Source	Specifies the message source address.
SourceRef	Specifies the source of the corresponding request message. This element takes the value of the request message Source element and is returned in the Status or Results element.
Target	Specifies the address of the node, in the DM Tree, that is the target of the OMA DM command.
TargetRef	Specifies the target address in the corresponding request message. This element takes the value of the request message Target element and is returned in the Status or Results element.
VerDTD	Specifies the major and minor version identifier of the OMA DM representation protocol specification used to represent the message.
VerProto	Specifies the major and minor version identifier of the OMA DM protocol specification used with the message.

Device management session

A device management (DM) session consists of a series of commands exchanged between a DM server and a phone. The server sends commands indicating operations that must be performed on the phone's management tree. The phone responds by sending commands that contain the results and any requested status information.

An example of a short DM session would be the following:

A server sends a Get command to a phone to retrieve the contents of one of the nodes of the management tree. The phone performs the operation and responds with a Result command that contains the requested contents.

A DM session can be divided into two phases:

- Setup phase: In response to a trigger event, a phone sends an initiating message to a DM server. The phone and server exchange needed authentication and phone information. This phase is represented by steps 1, 2, and 3 in the following table.
- Management phase: The DM server is in control. It sends management commands to the phone, and the phone responds. Phase two ends when the DM server stops sending commands and terminates the session. This phase is represented by steps 3, 4, and 5 in the following table.

The following table shows the sequence of events during a typical DM session.

Note 1: The step numbers in the table do not represent message identification numbers (MsgID). All messages from the server must have a MsgID that is unique within the session, starting at 1 for the first message and increasing by an increment of 1 for each additional message. For more

information about MsgID and OMA SyncML protocol, see "OMA Device Management Representation Protocol" (OMA-TS-DM_RepPro-V1_2-20070209-A) available from the OMA website.

Note 2: During OMA DM application level mutual authentication, if the device response code to Cred element in the server request is 212, no further authentication is needed for the remainder of the DM session. In the case of the MD5 authentication, the Chal element can however be returned. Then the next nonce in Chal MUST used for the MD5 digest when the next DM session is started.

If a request includes credentials and the response code to the request is 200, the same credential must be sent within the next request. If the Chal element is included and the MD5 authentication is required, a new digest is created by using the next nonce via Chal element for next request.

For more information about Basic or MD5 client authentication, MD5 server authentication, MD5 hash, and MD5 nonce, see the OMA Device Management Security specification (OMA-TS-DM_Security-V1_2_1-20080617-A),authentication response code handling and step-by-step samples in OMA Device Management Protocol specification (OMA-TS-DM_Protocol-V1_2_1-20080617-A) , available from the OMA website.

Step	Action	Description
1	The phone task schedule invokes the DM client.	At the scheduled time, the DM client is invoked periodically to call back to the enterprise management server over HTTPS.
2	The phone sends a message, over an IP connection, to initiate the session.	This message includes phone information and credentials. The client and server do certificate-based authentication over an SSL channel.
3	The DM server responds, over an IP connection (HTTPS).	The server sends initial device management commands, if any.
4	The phone responds to server management commands.	This message includes the results of performing the specified device management operations.
5	The DM server terminates the session or sends another command.	The DM session ends, or step 4 is repeated.

OMA DM provisioning files

OMA DM commands are transmitted between the server and the phone in messages. A message can contain one or more commands. For a list of commands supported in Windows Phone, see the table in OMA DM standards.

A DM message is an XML document. The structure and content of the document is defined in the OMA DM Representation Protocol (OMA-SyncML-DM_RepPro-V1_2_1-20080617-A.pdf) available from the OMA website.

Each message is composed of a header, specified by the SyncHdr element, and a message body, specified by the SyncBody element.

The following table shows the OMA DM versions that are supported in Windows Phone 8.

Version	Format
OMA DM version 1.1.2	`<SyncML xmlns='SYNCML:SYNCML1.1'> </SyncML>`
OMA DM version 1.2	`<SyncML xmlns='SYNCML:SYNCML1.2'> </SyncML>`

File format

The following example shows the general structure of the XML document sent by the server, using OMA DM version 1.2.1 for demonstration purposes only. The initial XML packages exchanged between client and server could contain additional XML tags. For a detailed description and samples for those packages, see the OMA Device Management Protocol 1.2.1 specification.

NOTE: XML encoding tag (<?xml version="1.0" encoding="UTF-8"?>) should not be included in the XML message.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>1</MsgID>
      <Target>
         <LocURI>{unique device ID}</LocURI>
      </Target>
      <Source>
         <LocURI>https://www.thephone-company.com/mgmt-server</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <!-- query a device OS system version -->
      <Get>
         <CmdID>2</CmdID>
         <Item>
            <Target>
               <LocURI>./DevDetail/SwV</LocURI>
            </Target>
         </Item>
      </Get>
      <!-- Update device policy -->

      <Final />
   </SyncBody>
</SyncML>
```

For more information about the header and body, see SyncHdr and SyncBody on MSDN.

SyncHdr element

SyncHdr includes the following information:

- Document Type Definition (DTD) and protocol version numbers
- Session and message identifiers. Note that each message in the same DM session must have a different MsgID.
- Message source and destination Uniform Resource Identifiers (URIs)
- Credentials for authentication

This information is used to by the phone to properly manage the DM session.

Code example

The following example shows the header component of a DM message. In this case, OMA DM version 1.2 is used as an example only.

> Note: The <LocURI> node value for the <Source> element in the SyncHdr of the phone-generated DM package should be the same as the value of ./DevInfo/DevID. For more information about DevID, see DevInfo configuration service provider.

```
<SyncHdr>
   <VerDTD>1.2</VerDTD>
   <VerProto>DM/1.2</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>{unique device ID}</LocURI>
   </Target>
   <Source>
      <LocURI>https://www.thephone-company.com/mgmt-server</LocURI>
   </Source>
</SyncHdr>
```

SyncBody element

SyncBody contains one or more DM commands.

Note that SyncBody can contain multiple DM commands; each command must have a different CmdID value.

Code example

The following example shows the body component of a DM message. In this example, SyncBody contains only one command, Get. This is indicated by the <Final /> tag that occurs immediately after the terminating tag for the Get command.

```
<SyncBody>
   <!-- query device OS software version -->
   <Get>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>./DevDetail/SwV</LocURI>
         </Target>
      </Item>
   </Get>
   <Final />
</SyncBody>
```

> Note: When using SyncML for OMA DM provisioning, a LocURI in SyncBody can have a "." as a valid segment name only in the first segment. However, a "." is not a valid segment name for the other segments. For example, the following LocURI is not valid because the segment name of the seventh segment is a ".".

```
<LocURI>./Vendor/MSFT/Registry/HKLM/System/./Test</LocURI>
```

Update phone settings example

The Replace command is used to update a phone setting.

Code example

The following example illustrates how to use the Replace command to update a phone setting.

```
<SyncHdr>
   <VerDTD>1.2</VerDTD>
   <VerProto>DM/1.2</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>{unique device ID}</LocURI>
   </Target>
   <Source>
      <LocURI>https://www.thephone-company.com/mgmt-server</LocURI>
   </Source>
</SyncHdr>
<SyncBody>
   <!-- update device setting -->
   <Replace>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>./Vendor/MSFT/PolicyManager/My/DeviceLock/MinDevicePasswordLength</LocURI>
         </Target>
         <Meta>
            <Type xmlns="syncml:metinf">text/plain</Type>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>6</Data>
      </Item>
   </Replace>
   <Final />
</SyncBody>
```

Server requirements for OMA DM

The following are the general server requirements for using OMA DM to manage Windows Phones:

- The OMA DM server must support the OMA DM v1.1.2 or later protocol.
- Secure Sockets Layer (SSL) must be on the OMA DM server, and it must provide server certificate-based authentication, data integrity checking, and data encryption. If the certificate is not issued by a commercial certification authority whose root certificate is preinstalled in the phone, you must provision the company's root certificate in the phone's ROOT store.
- To authenticate the client, you must use either Basic or MD5 client authentication at the application level. At the SSL level, use client certificate-based authentication.
- The server MD5 nonce must be renewed in each DM session for the next DM session. The DM client sends the new server nonce for the next session to the server by using the Status element in every DM session.
- The MD5 binary nonce is sent over XML in B64 encoded format, But the octal form of the binary data should be used when the server calculates the hash.

For more information about Basic or MD5 client authentication, MD5 hash generation, and MD5 nonce, see the OMA Device Management Security specification (OMA-TS-DM_Security-V1_2_1-20080617-A) and OMA Device Management Protocol specification (OMA-TS-DM_Protocol-V1_2_1-20080617-A), available from the OMA website.

Enterprise OMA DM supported configuration service providers

The following configuration service providers are supported via OMA DM. See Configuration service provider reference for detailed descriptions of each configuration service provider.

- CertificateStore configuration service provider – updated in Windows Phone 8.1
- DMS configuration service provider
- DMClient configuration service provider – updated in Windows Phone 8.1
- EnterpriseAppManagement configuration service provider
- DeviceLock configuration service provider – superseded by PolicyManager configuration service provider
- Storage configuration service provider - superseded by PolicyManager configuration service provider
- DevInfo configuration service provider
- DevDetail configuration service provider
- RemoteWipe configuration service provider
- Email2 configuration service provider
- ActiveSync configuration service provider
- NodeCache configuration service provider
- Phone encryption via PolicyManager CSP
- PolicyManager configuration service provider – new in Windows Phone 8.1
- RemoteLock configuration service provider – new in Windows Phone 8.1
- Wi-Fi configuration service provider – new in Windows Phone 8.1
- VPN configuration service provider – new in Windows Phone 8.1
- RemoteRing configuration service provider – new in Windows Phone 8.1
-

DM client configuration (Updated in Windows Phone 8.1)

The W7 APPLICATION CSP is built based on the standard OMA Client Provisioning W7 APPLICATION characteristic definition. This CSP allows the enrollment server to configure the DM client to communicate with the server. Windows Phone has extended it with a few more parameters to provide richer configuration capability. For more information, see w7 APPLICATION configuration service provider later in this document.

Windows Phone 8.1 allows the enrollment server to specify how frequently the DM client should call back to the management server. This is configured by the server sending OMA Client Provisioning XML via the DMClient CSP.

Note that in Windows Phone 8, the enrollment server configures scheduled DM events (pulling MDM server, start MDM client certificate manual renew process) via the Registry CSP during the enrollment process. These registry keys are only set and used during the enrollment process. The usage of those registry keys are deprecated in post Windows Phone 8, and instead, starting with Windows Phone 8.1, the DMClient CSP is used to configure scheduled DM polling events during enrollment. Those DM polling schedules should be updated later by the MDM server via the DMClient CSP. The behavior of those polling parameters are also updated. Please refer to the DMClient configuration service provider section for a detailed description.

- The following example shows how to configure a DM client by using W7 APPLICATION provisioning XML sent during enrollment.

```
<wap-provisioningdoc version="1.1">
   <characteristic type="APPLICATION">
```

```
      <parm name="APPID" value="w7"/>
      <parm name="PROVIDER-ID" value="TestMDMServer"/>
      <parm name="NAME" value="Microsoft"/>
      <parm name="ADDR" value="https://DM.contoso.com:443/omadm/WindowsPhone.ashx"/>
      <parm name="CONNRETRYFREQ" value="6" />
      <parm name="INITIALBACKOFFTIME" value="30000" />
      <parm name="MAXBACKOFFTIME" value="120000" />
      <parm name="BACKCOMPATRETRYDISABLED" />
      <parm name="DEFAULTENCODING" value="application/vnd.syncml.dm+wbxml" />
      <parm name="SSLCLIENTCERTSEARCHCRITERIA" value=
"Subject=DC%3dcom%2cDC%3dmicrosoft%2cCN%3dUsers%2cCN%3dAdministrator&amp;Stores=My%5CUser"/>
      <characteristic type="APPAUTH">
         <parm name="AAUTHLEVEL" value="CLIENT"/>
         <parm name="AAUTHTYPE" value="DIGEST"/>
         <parm name="AAUTHSECRET" value="password1"/>
         <parm name="AAUTHDATA" value="B64encodedBinaryNonceInsertedHere"/>
      </characteristic>
      <characteristic type="APPAUTH">
         <parm name="AAUTHLEVEL" value="APPSRV"/>
         <parm name="AAUTHTYPE" value="BASIC"/>
         <parm name="AAUTHNAME" value="testclient"/>
         <parm name="AAUTHSECRET" value="password2"/>
      </characteristic>
   </characteristic>
</wap-provisioningdoc>
```

NOTE 1: parm name and characteristic type in w7 APPLICATION CSP XML are case sensitive and must be all uppercase.

NOTE 2: In w7 APPLICATION characteristics, both CLIENT and APPSRV credentials should be provided in XML. For detailed description, refer w7 APPLICATION configuration service provider.

Enterprise-specific DM client configuration

The DMClient configuration service provider is used to specify additional enterprise-specific configuration settings for identifying the phone in the enterprise domain, security mitigation for certificate renewal, and server-triggered enterprise unenrollment. For more information, see DMClient configuration service provider later in this document.

DM Client Push Support in Windows Phone 8.1, DMClient supports the ability to configure Push initiated device management sessions. Utilizing Windows Notification Service (WNS), a management server can request a device to establish a management session with the server through a push notification. Once a device is configured to support Push by the management server by providing the device with a PFN, the device will register a persistent connection with the WNS cloud (Battery Sense and Data Sense conditions permitting).

In order to initiate a device management session, the management server must first authenticate with WNS using its SID and client secret. Once authenticated, the server will receive a token that it can use to initiate a raw push notification for any ChannelURI. When the management server wishes to initiate a device management session with a device, it can utilize its token and the device's ChannelURI and begin communicating with the device.

Because a device may not be currently connected to the WNS cloud, it is possible to configure the raw notification request to get status information back from the WNS cloud. The server can receive the connection status when it sends a push notification using a device's ChannelURI using the X-WNS-

RequestForStatus header. This will instruct WNS to return to the server whether or not a device is connected to WNS. This can be used by the management server to determine if a push notification has reached the device. Additionally, if the server wishes to send a time-bound raw push notification, the server can use the X-WNS-TTL header that will provide WNS with a time-to-live binding so that the notification will expire after the time has passed. Please see this MSDN article for more details on Push notification service request and response headers.

For more information and sample code to build the server-side components that initiate raw push notifications, please see these MSDN articles on Push notification overview and Sending a push notification.

Please note the following restrictions as related to push notifications and WNS

- All push notifications are delivered by "best effort." The notification is not guaranteed to be delivered to the device.
- Push for device management uses raw push notifications. This means that these raw push notifications do not support or utilize push notification payloads.
- Each ChannelURI has a limit of 150 push notifications per hour
- Receipt of push Notifications are sensitive to the Battery Sense and Data Sense settings on the device. For example, if the battery drops below certain thresholds, the device's persistent connection with WNS will be terminated. Additionally, if the user is utilizing Data Sense and has exceeded their monthly altoment of data, the device's persistent connection with WNS will also be terminated.
- A ChannelURI provided to the management server by the device is only valid for 30 days, and can be revoked prior to the lapse of the 30 days. The device will automatically renew the ChannelURI after 15 days and trigger a management session on successful renewal of the ChannelURI. It is **strongly** recommended that, during every management session, the management server queries the ChannelURI value to ensure that it has received the latest value. This will ensure that the management server will not attempt to use a ChannelURI that has expired.
- Push is not a replacement for having a polling schedules
- PFNs can be revoked by WNS if improper use of PFNs and Push is detected. Any devices being managed using this PFN will cease to have Push initiated device management support.

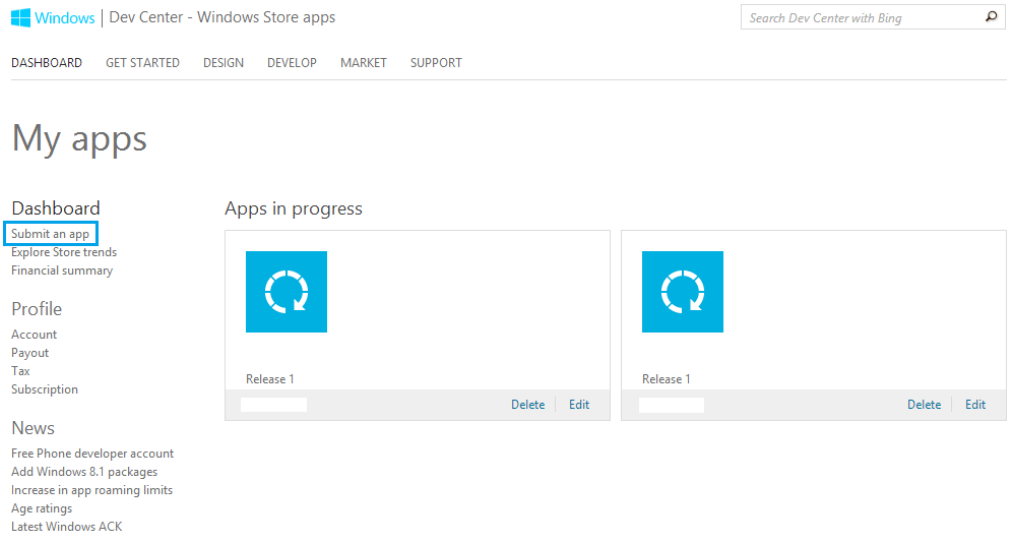
Please note that Push does not support configuration using WAP Provisioning XML and cannot be configured during an enrollment session. Configuration of Push should occur during a management session.

Getting Push credentials through Windows Store

For utilizing Push in production please follow the steps for creating a raw notification listed at http://msdn.microsoft.com/en-us/library/windows/apps/xaml/jj676791.aspx.

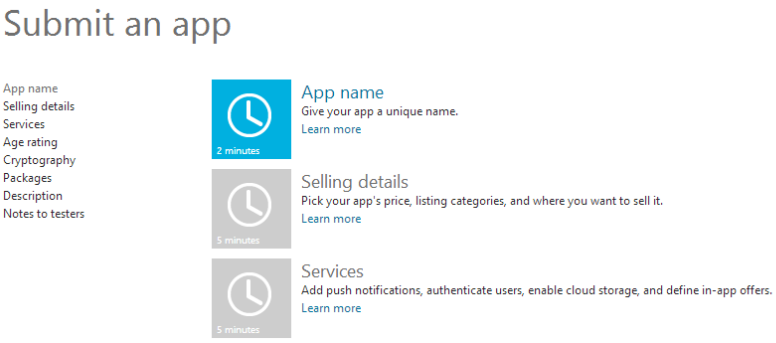
Acquiring application-based WNS credentials for MDM Push

Start by visiting https://appdev.microsoft.com/StorePortals/en-us/Home/Index and sign in with your MSDN developer account

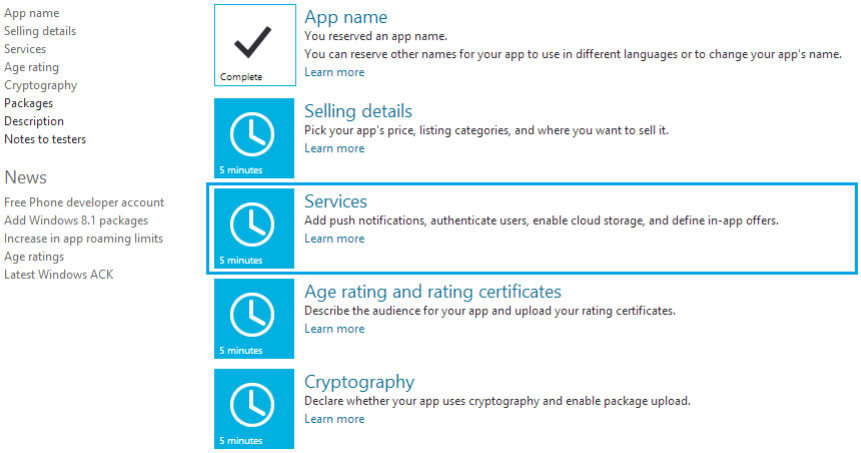


Create a Windows application with a reserved application name.

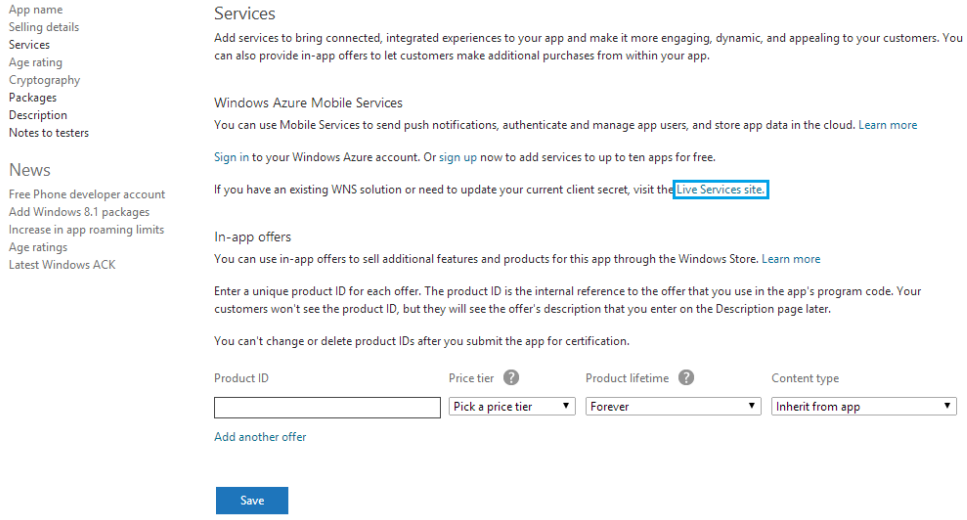
Make sure you never delete this application, or your push notification credentials will be invalidated.



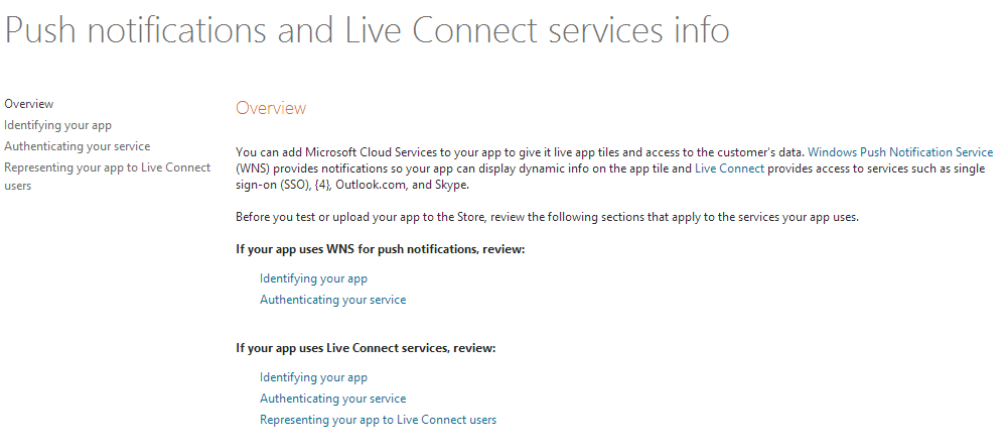
Once the application name is reserved, click on “Services”



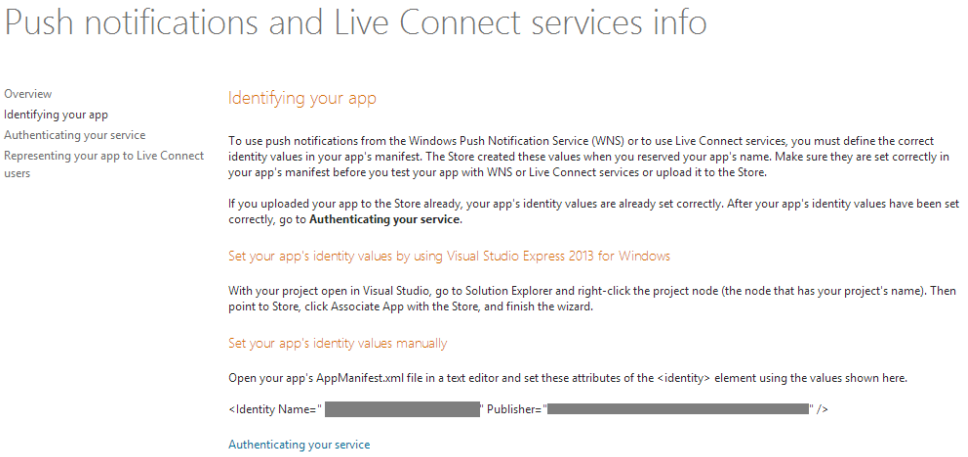
Click on the "Live Services" link



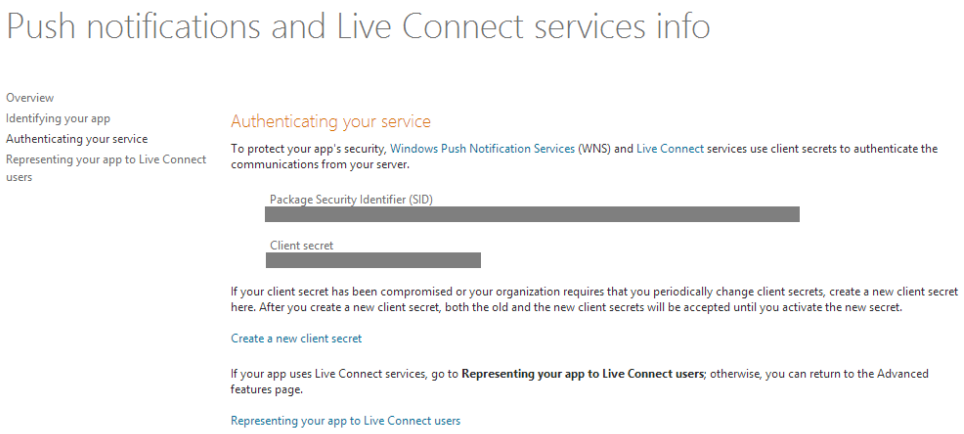
This will re-direct to the following page:



Click on "Identifying your application". Record the <Identity /> portion. This will be used inside a temporary Windows application



The following information will be required in order to authenticate the server sending a raw Push notification request to the device once you <Get> the ChannelURI from the device

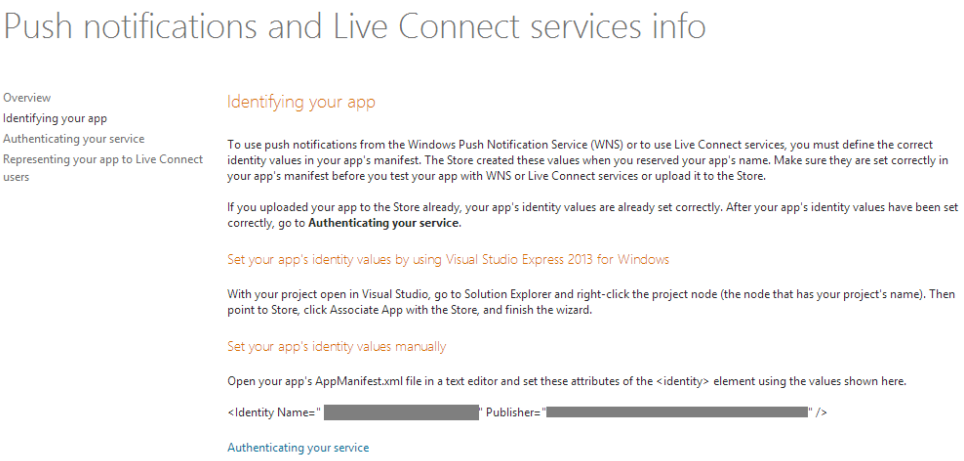


For more information on sending Push notifications, please visit this site.

Generate your PFN

Open Visual Studio and create a new solution for a Windows Store application. You may be prompted to sign into your MSDN developer account.

In the Package.appmanifest file, modify the <Identity /> field to include the Name and Publisher from above from "Set your app's identity values manually"



In Visual Studio, select **Build** -> **Deploy Solution**. If you previously did not get your Developer License, you may be prompted to do so again at this point.

In the console output, there will be a reference to the Package Full Name. This is the superset of the PFN that you should send to the device

```
------ Build started: Project: App1, Configuration: Debug Any CPU ------
```

```
  App1 -> c:\users\<username>\documents\visual studio
2013\Projects\App1\App1\bin\Debug\App1.exe
------ Deploy started: Project: App1, Configuration: Debug Any CPU ------
Creating a new clean layout...
Copying files: Total <1 mb to layout...
Registering the application to run from layout...
Deployment complete. Full package name: "THIS_IS_YOUR_PFN"
```

The PFN should come in some format like:

```
XXXXXXXXXXXX.NameOfApp_1.1.1.1_neutral_xxxxxxxxxxxx
```

Please remove the section with the app version and instead just use this portion in this format

```
XXXXXXXXXXXX.NameOfApp_xxxxxxxxxxxx
```

Please see the section below on DMClient configuration provider for setting the PFN on the device.

Enterprise app management over DM server

The EnterpriseAppManagement configuration service provider is used to handle enterprise application management tasks such as installing an enterprise application token, the first auto-downloadable app link, querying installed enterprise applications (name and version), auto updating already installed enterprise applications, and removing all installed enterprise apps (including the enterprise app token) during unenrollment. For more information and sample commands, see EnterpriseAppManagement configuration service provider later in this document.

Enterprise application install, update, uninstall (Update in Windows Phone 8.1)

As part of the EnterpriseAppManagment CSP, management servers have the ability to install, update, and uninstall Line of Business applications during a management session. This behavior is supported for all application file format types including XAP, AppX, and AppXBundle.

Note that you cannot install company hub apps to an SD card. This is not supported in Windows Phone 8.1.

Enterprise application restrictions (New in Windows Phone 8.1)

As part of the PolicyManager CSP, management servers have the ability to configure a list of applications or set of applications from publishers that can be allowed or denied. This solution provides full flexibility to block both 3rd party applications from the Windows Phone Store and Line of Business applications on the device. There is no support for blocking native 1st party applications published by Microsoft through application restrictions. The two exceptions include Internet Explorer and Store, which can be blocked through the PolicyManager CSP directly. There are two pre-defined sets of XML schemas that are utilize for defining application restrictions. Please note that previously running applications in the back stack may not be immediately terminated upon successful setting of the ApplicationRestrictions policy.

Application Restrictions - Allow List

An allow list contains a set of applications defined by a set of application GUIDs and application publisher names that are allowed to be installed and run on the device. Any applications that are not explicitly

listed or published underneath an allowed publisher will not available for use or install by the user. Here is a sample:

```
<Data>
&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;
&#x3C;Allow&#x3E;
  &#x3C;!-- Allow App01 with a WindowsPhone.com GUID of {f5f53dbf-c7bd-4b26-a1bf-e1cf8d69b9d5}
--&#x3E;
  &#x3C;App ProductId=&#x22;{f5f53dbf-c7bd-4b26-a1bf-e1cf8d69b9d5}&#x22;/&#x3E;
  &#x3C;!-- Allow Publisher Contoso --&#x3E;
  &#x3C;Publisher PublisherName=&#x22;Contoso&#x22; /&#x3E;
  &#x3C;!-- Allow Publisher Fabrikam --&#x3E;
  &#x3C;Publisher PublisherName=&#x22;Fabrikam&#x22;&#x3E;
    &#x3C;!-- Deny FabrikamApp01 with a WindowsPhone.com GUID of {b79fb25e-ea4a-4dda-bbba-
66c282377105} --&#x3E;
    &#x3C;DenyApp ProductId=&#x22;{b79fb25e-ea4a-4dda-bbba-66c282377105}&#x22;/&#x3E;
  &#x3C;/Publisher&#x3E;
&#x3C;/Allow&#x3E;&#x9;
&#x3C;/AppPolicy&#x3E;</Data>
```

In this example, application App01 and any applications published by Contoso will be allowed to be installed and run. Additionally, all applications published by Fabrikam except for FabrikamApp01 can also be installed and run. All other Line of Business applications and Windows Phone Store applications cannot be run.

For the latest XML schema definition for Application Restrictions XML, please visit http://schemas.microsoft.com/phone/2013/policy

Application Restrictions - Deny List

A deny list contains a set of applications defined by a set of application GUIDs and application publisher names that cannot to be installed or run on the device if the application already exists on the device. Here is a sample:

```
<Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;
&#x3C;Deny&#x3E;
  &#x3C;!-- Deny App01 with a WindowsPhone.com GUID of {f5f53dbf-c7bd-4b26-a1bf-e1cf8d69b9d5}
--&#x3E;
  &#x3C;App ProductId=&#x22;{f5f53dbf-c7bd-4b26-a1bf-e1cf8d69b9d5}&#x22; /&#x3E;
  &#x3C;!-- Deny Publisher Contoso --&#x3E;
  &#x3C;Publisher PublisherName=&#x22;Contoso&#x22; /&#x3E;
  &#x3C;!-- Deny Publisher Fabrikam --&#x3E;
  &#x3C;Publisher PublisherName=&#x22;Fabrikam&#x22;&#x3E;
    &#x3C;!-- Allow FabrikamApp01 with a WindowsPhone.com GUID of {b79fb25e-ea4a-4dda-bbba-
66c282377105} --&#x3E;
    &#x3C;AllowApp ProductId=&#x22;{b79fb25e-ea4a-4dda-bbba-66c282377105}&#x22; /&#x3E;
  &#x3C;/Publisher&#x3E;
&#x3C;/Deny&#x3E;
&#x3C;/AppPolicy&#x3E;
</Data>
```

In this sample, application App01 and any applications published by Fabrikam cannot be installed or run on the device. Additionally, any applications published by Fabrikam except for FabrikamApp01 cannot be installed or run on the device. All other applications can be installed and run on the device.

For the latest XML schema definition for Application Restrictions XML, please visit http://schemas.microsoft.com/phone/2013/policy. ApplicationRestrictions XSD is also provided in the Appendix

Guide to debugging allow/deny lists XMLs

A number of allow and deny list samples are provided in the Appendix for easy reference.

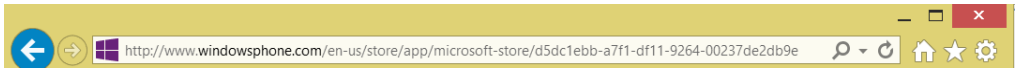
When debugging why an allow/deny list, please ensure the following:

- Remove all line feeds, return characters, carriage returns in the <Data> payload. Additionally, Ensure that the beginning of the <Data> field always is followed by <![CDATA[without a return or line feed in between. This may cause schema validation to reject the payload.
- Ensure an <Atomic> tag is used to wrap the <Replace>. It isn't strictly required, but it is the recommended and test approach to successfully configuring Application Restrictions
- Ensure that <AllowApp> is only used within a <Deny> List following a <Publisher> and proceeding a </Publisher> Tag
- ProductId is case sensitive. Ensure that the "d" is lowercase
- The value of ProductId is case sensitive. Ensure that characters are all in lowercase.
- Always wrap ProductIds in curly braces. For example, ProductId="{<product-id>}"
- The PublisherName includes all punctuation. WindowsPhone.com may omit punctuation from being displayed on the website. As a result, some publishers will have multiple PublisherNames that should be blocked

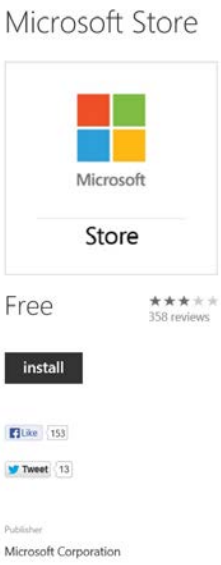
Finding application GUIDs or application publisher name

The primary way for finding an application GUID or application publisher name is through www.windowsphone.com. In this example, the Microsoft Store application is being represented as an application that the admin would like to use

The application GUIDs are listed as part of the URL when an IT administrator finds an application they would like to include in their application restrictions list. In this case, the Microsoft Store application has an application GUID of **d5dc1ebb-a7f1-df11-9264-00237de2db9e**.



The main window also shows the publisher for this application as **Microsoft Corporation**



For more information on how to send down Application Restrictions XMLs, please see the samples in the PolicyManager CSP later in this document. Additional samples are provided in Appendix.

Device lock policy configuration

The DeviceLock configuration service provider allows the management server to configure device lock related policies. The policies configured via this CSP are superseded by Windows Phone 8.1 new CSP – PolicyManager CSP which not only configures device lock related policies but other Windows Phone 8 and Windows Phone 8.1 enterprise policies.

Note: DeviceLock CSP will be deprecated post Windows Phone 8.1. It is recommended that MDM server should use PolicyManager CSP to configure device lock policies for Windows Phone 8.1 device.

In Windows Phone, to help safeguard device policies from being compromised by an untrusted authority, the phone builds in the most secure logic for device lock polices. For example, if both Exchange and the management server push device lock policies to the phone, the phone applies the most secure policy value. If an account is removed from the phone, the next most secure policy value set by the remaining accounts is applied. For more information and sample commands, see PolicyManager configuration service provider.

Encryption

Windows Phone supports internal storage encryption. The enterprise management server can enable the encryption. The removable storage card is not encrypted.

Note that after encryption is enabled, it cannot be disabled. The Storage UI is the visual indicator of the encryption state. Only those phones that have UEFI Secure Boot enabled support device encryption. The emulator doesn't support device encryption.

- Note that in Windows Phone 8.1 the PolicyManager csp is used to set and query device encryption status.

Querying device encryption status

The server can query PolicyManager/Device/Security/RequireDeviceEncryption node value to find out whether the phone is encrypted:

The following XML sample shows how to query encryption state via SyncXML command.

```
<Get>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/PolicyManager/Device/Security/RequireDeviceEncryption
         </LocURI>
      </Target>
   </Item>
</Get>
```

Enabling internal storage encryption

To enable internal storage encryption, the enterprise management server can set the following PolicyManager/My/Security/RequireDeviceEncryption value

Note: the emulator does not support encryption.

- The following sample shows how to enable internal storage encryption via SyncML XML command.

```
<Replace>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/PolicyManager/My/Security/RequireDeviceEncryption
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>1</Data>
   </Item>
</Replace>
```

Remote wipe

The RemoteWipe configuration service provider is used by the server to remotely wipe the phone. The phone's internal user storage is wiped, including all user's personal content as apps, emails, contacts, media files, if any. Consequently, the phone also loses the information about connecting to the enterprise. For more information, see RemoteWipe configuration service provider later in this document.

Storage card policy configuration

In Windows Phone 8.1, the PolicyManager configuration service provider's My/Security/AllowStorageCard policy allows the management server to remotely disable or enable the storage card itself. For more information, see later in this document. Note that in Windows Phone 8, the Storage configuration service

provider is used to configure storage card policy. While this is still supported Windows Phone 8.1, it will be depreciated post Windows Phone 8.1. It is recommended MDM server uses PolicyManager CSP to configure company policies including storage card policy starting in Windows Phone 8.1.

In addition to noting the DM command response from the phone, the server can query the PolicyManager CSP to confirm that the policy has been applied.

The following SyncML sample shows how to disable the storage card by using SyncML.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/My/System/AllowStorageCard </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>0</Data>
   </Item>
</Replace>
```

Cellular app download limit configuration (new for GDR2)

In Windows Phone 8.1 GDR2, the Connectivity/CellularAppDownloadMBLimit policy is added that blocks cellular application download if the application file size exceeds the specified file limit. It prevents excessive celluar data cost for large application downloads when Wi-Fi connection is not available.

The following SyncML sample shows how to set the default cellular download limit to 20MB for app download.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/Device/Connectivity/CellularAppDownloadMBLimit</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>0</Data>
   </Item>
</Replace>
```

Data protection under lock (new for GDR2)

In Windows Phone 8.1 GDR2, for organizations looking to protect themselves from enterprise data leakage when device is PIN locked and a device gets lost or stolen, an additional policy can be applied by MDM that will help protect access to enterprise email and attachments when device is PIN locked. Protected email accounts can be provisioned either via EAS or MDM. In addition, the protected data will be encrypted (using a separate enterprise key) at all times. This policy is in addition to the existing device encryption policy that applies globally to all data (personal and business) on the device.

While the GDR2 policy for data protection under lock applies only to enterprise emails and associated attachments, in future releases, this same policy will expand its scope to other types of enterprise content. Future versions of this document will document support for additional applications as they get released.

The DataProtection/RequireProtectionUnderLockConfig policy allows data encryption of email data and associated attachments. Pin lock key is required to unlock and decode the content. In addition to the policy for RequireProtectionUnderLockConfig, the management server must also define a list of email domains associated with their enterprise email infrastructure. This will ensure that the protection policy only applies to corporate accounts and private email accounts will not be affected.

Note that this policy breaks the conversation view on EAS version older that 14.0. If you want to use this policy on EAS versions older than 14.0, then you should disable the conversation view.

The following SyncML sample shows how to enable email data protection under Lock.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/DataProtection/RequireProtectionUnderLockConfig</LocURI
>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>0</Data>
   </Item>
</Replace>
```

The following SyncML sample shows how to provision the required "protected domains" that will define the set of accounts on the device that would be protected under lock.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/DataProtection/EnterpriseProtectedDomainNames</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
         <Type xmlns=”syncml:metinf”>text/plain</Type>
      </Meta>
      <Data>email*.contoso.com</Data>
   </Item>
</Replace>
```

Enterprise anti-theft override (new for GDR2)

In Windows Phone GDR2, The Security/AntiTheftMode policy is added to allow the enterprise to disable the anti-theft mode of Windows Phone devices by overwriting policy when user does not have it turned on. It prevents enterprise managed devices from being locked with individual user. Note that the policy does not enable anti theft roll back. User will still have to manually disable the anti-theft mode before this policy could take effect.

The following SyncML sample shows how to disable Anti Theft mode.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/My/Security/AntiTheftMode</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>1</Data>
   </Item>
</Replace>
```

Fully managed VPN setting (new for GDR2)

In Windows Phone GDR2, the Connectivity/AllowManualVPNConfiguration policy is added to prevent user from creating/changing VPN profiles or toggle VPN off. It prevents users from circumventing enterprise security policy for data in transit.

The following SyncML sample shows how to enable fully managed VPN setting.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/Connectivity/AllowManualVPNConfiguration</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>1</Data>
   </Item>
</Replace>
```

Task switcher control (new for GDR2)

In Windows Phone 8.1 GDR2, the Device/Experience/AllowTaskSwitcher policy is added to allow an enterprise that is concerned about data leak to prevent the user from using the task switcher. Note that it does not effect the normal back button function because only the visual switcher is disabled.

The following SyncML sample shows how to disable Task Switcher.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/My/Experience/AllowTaskSwitcher</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>1</Data>
```

```
   </Item>
</Replace>
```

WLAN scan frequency customization (new for GDR2)

In Windows Phone 8.1 GDR2, the Connectivity/WLANScanMode policy is added to allow an enterprise to control the WLAN scanning behavior and how aggressively devices should be actively scanning for Wi-Fi networks to get devices connected.

The following SyncML sample shows how to customize the WLAN scan frequency.

```
<Replace>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/My/Wifi/WLANScanMode</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>0</Data>
   </Item>
</Replace>
```

Bulk enrollment (new for GDR2)

The IT administrator can provision Windows Phone devices using an SD card or a USB tether MTP file transfer. First you create a customization file and then load it the phone using an SD card or copy it to the phone through a USB connection. Windows Phone detects the customization file from the OOBE start up.

Here's the list of customizations you can configure:

- Add a certificate file
- Add a Wi-Fi profile
- Set the system time server
- Set the system time zone
- Set the langua and locale
- Set the OOBE configuration
- Set the MDM server setting

Apply the customization using a USB connection to the phone

1. Boot the Windows Phone into the first screen of the OOBE.
2. Connect the phone to the PC using a USB. The PC should automatically detect the Windows Phone and show the File Explorer.
3. Copy the customizations.xml file to the root folder of the phone. The phone automatically detects the customization file and shows a confirmation page.
4. Click **Done**.
5. Remove the USB connection.

Apply the customization using an SD card

1. Copy the customization.xm file to the root folder of the SD card.
2. Insert the SD card into the Windows Phone.
3. Boot the Windows Phone into the OOBE start screen.
4. The phone automatically detects the customization file and shows a confirmation page. Click **Done**.

After the customization is applied to the phone, you can remove the SD card.

Add a certificate file

Here's an example:

```
<!-- Certificate Store CSP -->
      <CertificateStore>
        <CA>
          <System ThumbPrint="92F6A5FF349A519F26C8D863758904380FB97F97">
              <EncodedCertificate> EncodedCertificate>
          </System>
        </CA>
      </CertificateStore>
```

The **System ThumbPrint** is the actual thumbprint for the certificate. The **EncodedCertificate** contains the Base-64 encoded x.509 certificate.

You can export the certificate through the certmgr app from the Windows Control Panel using one of the following methods:

Certificate export method 1

1. Select a certificate and then right-click.
2. Select **All Tasks** > **Export**.
3. Select **Export File format Base-64 encoded x.509**
4. Copy the encoded certificate to the customizations.xml file.

Certificate export method 2

1. Select a certificate and then right-click.
2. Select **All Tasks** > **Export**.
3. Select Export File format DER encoded binary.
4. Use the WEH 8.1 Prov Encryption/BASE64 Tool from CodePlex to convert it to Base 64.
5. Copy the encoded certificate to the customizations.xml file.

For additional information, see CryptHashCertificate and X509Certificate2.Thumbprint.

Add a Wi-Fi profile

Here's an example:

```
<!-- Wifi CSP -->
      <WiFi>
        <Profile name="Wifi_Contoso">
            <WlanXml>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;Wifi_Chan
dler&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;Wifi_Chandler&lt;/name&gt;&lt;/SSID
```

```
&gt;&lt;nonBroadcast&gt;true&lt;/nonBroadcast&gt;&lt;/SSIDConfig&gt;&lt;connectionType&gt;ESS&
lt;/connectionType&gt;&lt;connectionMode&gt;auto&lt;/connectionMode&gt;&lt;MSM&gt;&lt;security
&gt;&lt;authEncryption&gt;&lt;authentication&gt;WPA2PSK&lt;/authentication&gt;&lt;encryption&g
t;AES&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;sharedKey&gt;&lt;keyType&gt;passPhrase&lt;/
keyType&gt;&lt;protected&gt;false&lt;/protected&gt;&lt;keyMaterial&gt;Microsoft1234&lt;/keyMat
erial&gt;&lt;/sharedKey&gt;&lt;/security&gt;&lt;/MSM&gt;&lt;/WLANProfile&gt;</WlanXml>
        </Profile>
      </WiFi>
```

The **Profile name** is the actual profile name.

To export an existing Wi-Fi profile

1. Run **cmd** as an administrator.
2. Run **netsh**.
3. Run export profile folder=c:\profiles name="WiFi Profile name".
4. Copy the content of the exported profile as encoded text into **WlanXml** in the customizations.xml.

Set the system time server

Here's an example:

```
<!-- System Time CSP -->
      <MCSF>
        <AutomaticTime>
          <NTPRegularSyncInterval>1</NTPRegularSyncInterval>
          <NTPServers>time.windows.com&#xF000;time.nist.gov&#xF000;&#xF000;</NTPServers>
        </AutomaticTime>
      </MCSF>
```

To set the regular sync interval in hours, set **NTPRegularSyncInterval** to a value between 1 and 168 hours (inclusive). The default sync internal is 12 hours.

Set the system time zone

Here is an example for setting the time zone for India.

```
<!-- Time Zone Settings for India-->
      <AutomaticTime>
      <TimeZonePriority1>0x6B8</TimeZonePriority1>
      </AutomaticTime>
```

Time zone priority list:

```
ID     Time zone
=========================================
0x0    UTC-12 International Date Line West
0x6E   UTC-11 Coordinated Universal Time-11
0xC8   UTC-10 Hawaii
0x12C  UTC-09 Alaska
0x190  UTC-08 Pacific Time (US & Canada)
0x19A  UTC-08 Baja California
0x1F4  UTC-07 Mountain Time (US & Canada)
0x1FE  UTC-07 Chihuahua, La Paz, Mazatlan
0x208  UTC-07 Arizona
0x258  UTC-06 Saskatchewan
0x262  UTC-06 Central America
0x26C  UTC-06 Central Time (US & Canada)
```

```
0x276  UTC-06 Guadalajara, Mexico City, Monterrey
0x2BC  UTC-05 Eastern Time (US & Canada)
0x2C6  UTC-05 Bogota, Lima, Quito
0x2D0  UTC-05 Indiana (East)
0x320  UTC-04 Atlantic Time (Canada)
0x32A  UTC-04 Cuiaba
0x334  UTC-04 Santiago
0x33E  UTC-04 Georgetown, La Paz, Manaus, San Juan
0x348  UTC-04:30 Caracas
0x352  UTC-04 Asuncion
0x384  UTC-03:30 Newfoundland
0x38E  UTC-03 Brasilia
0x398  UTC-03 Greenland
0x3A2  UTC-03 Montevideo
0x3AC  UTC-03 Cayenne, Fortaleza
0x3B6  UTC-03 Buenos Aires
0x3C0  UTC-03 Salvador
0x3E8  UTC-02 Mid-Atlantic
0x3F2  UTC-02 Coordinated Universal Time-02
0x44C  UTC-01 Azores
0x456  UTC-01 Cabo Verde Is.
0x4B0  UTC Dublin, Edinburgh, Lisbon, London
0x4BA  UTC  Monrovia, Reykjavik
0x4C4  UTC  Casablanca
0x4CE  UTC  Coordinated Universal Time
0x514  UTC+01 Belgrade, Bratislava, Budapest, Ljubljana, Prague
0x51E  UTC+01 Sarajevo, Skopje, Warsaw, Zagreb
0x528  UTC+01 Brussels, Copenhagen, Madrid, Paris
0x532  UTC+01 West Central Africa
0x53C  UTC+01 Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
0x546  UTC+01 Windhoek
0x578  UTC+02 E. Europe
0x582  UTC+02 Cairo
0x58C  UTC+02 Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
0x596  UTC+02 Athens, Bucharest
0x5A0  UTC+02 Jerusalem
0x5AA  UTC+02 Amman
0x5B4  UTC+02 Beirut
0x5BE  UTC+02 Harare, Pretoria
0x5C8  UTC+02 Damascus
0x5D2  UTC+02 Istanbul
0x5DC  UTC+03 Kuwait, Riyadh
0x5E6  UTC+03 Baghdad
0x5F0  UTC+03 Nairobi
0x5FA  UTC+02 Kaliningrad (RTZ 1)
0x60E  UTC+03:30 Tehran
0x604  UTC+03 Moscow, St. Petersburg, Volgograd (RTZ 2)
0x640  UTC+04 Abu Dhabi, Muscat
0x618  UTC+03 Minsk
0x64A  UTC+04 Baku
0x654  UTC+04 Yerevan
0x65E  UTC+04:30 Kabul
0x668  UTC+04 Tbilisi
0x672  UTC+04 Port Louis
0x67C  UTC+04 Izhevsk, Samara (RTZ 3)
0x6AE  UTC+05 Tashkent
0x6B8  UTC+05:30 Chennai, Kolkata, Mumbai, New Delhi
0x6C2  UTC+05:30 Sri Jayawardenepura
0x6CC  UTC+05:45 Kathmandu
0x6D6  UTC+05 Islamabad, Karachi
0x6A4  UTC+05 Ekaterinburg (RTZ 4)
0x708  UTC+06 Astana
```

```
0x71C  UTC+06:30 Yangon (Rangoon)
0x726  UTC+06 Dhaka
0x712  UTC+06 Novosibirsk (RTZ 5)
0x776  UTC+07 Bangkok, Hanoi, Jakarta
0x76C  UTC+07 Krasnoyarsk (RTZ 6)
0x7D0  UTC+08 Beijing, Chongqing, Hong Kong SAR, Urumqi
0x7E4  UTC+08 Kuala Lumpur, Singapore
0x7EE  UTC+08 Taipei
0x7F8  UTC+08 Perth
0x802  UTC+08 Ulaanbaatar
0x7DA  UTC+08 Irkutsk (RTZ 7)
0x834  UTC+09 Seoul
0x83E  UTC+09 Osaka, Sapporo, Tokyo
0x852  UTC+09:30 Darwin
0x85C  UTC+09:30 Adelaide
0x848  UTC+09 Yakutsk (RTZ 8)
0x898  UTC+10 Canberra, Melbourne, Sydney
0x8A2  UTC+10 Brisbane
0x8AC  UTC+10 Hobart
0x8C0  UTC+10 Guam, Port Moresby
0x8B6  UTC+10 Vladivostok, Magadan (RTZ 9)
0x8FC  UTC+11 Solomon Is., New Caledonia
0x906  UTC+12 Magadan
0x91A  UTC+11 Chokurdakh (RTZ 10)
0x960  UTC+12 Fiji
0x96A  UTC+12 Auckland, Wellington
0x974  UTC+12 Petropavlovsk-Kamchatsky - Old
0x97E  UTC+12 Coordinated Universal Time+12
0x988  UTC+12 Anadyr, Petropavlovsk-Kamchatsky (RTZ 11)
0x9C4  UTC+13 Nuku'alofa
0x64   UTC+13 Samoa
0xA28  UTC+14 Kiritimati Island
```

Set the language and locale

Here is an example. For more information about the Windows Phone languages, see Phone_languages.

```
<!-- OOBE Settings -->
       <Oobe>
         <AcceptTermsOfUse>true</AcceptTermsOfUse>
         <SkipSettings>true</SkipSettings>
         <SkipOnlineConsumerRegistration>true</SkipOnlineConsumerRegistration>
      </Oobe>
```

When **AcceptTermsOfUse** is set to true, the **Accept Term of Use** screen will not be displayed to the user and default value is set.

When **SkipSetting** is set to true, the **Customize Setting** pages from OOBE are skipped and the default values are set.

When **SkipOnlineConsumerRegisteration** is set to true, the **Microsoft User Account** setting flow is skipped during OOBE.

Set the MDM server setting

Here's an example:

```
<!-- MDM Settings -->
      <EnterpriseExt>
        <MDM>
          <Server>enterpriseenrollment-s.manage-beta.microsoft.com</Server>
```

```
      </MDM>
     </EnterpriseExt>
   </Common>
  </Settings>
 </Customization>
```

When you set the MDM enrollment server, it triggers the MDM enrollment flow where the user may sign up to the MDM server. The IT administrator is required to provide a valid Wi-Fi profile to enable the network connection with a specific MDM server.

Sample customizations.xml

```
<Customization version="1.0">
  <Settings>
    <Common>
<!-- Certificate Store CSP -->
      <CertificateStore>
        <CA>
          <System ThumbPrint="7ca93a74e10fc99ca948c15802032f9c25c24abc">
              <EncodedCertificate>
MIIEqTCCA5GgAwIBAgITfQHmAwbWyZWKmX27SwAEAeYDBjANBgkqhkiG9w0BAQUF
ADAfMR0wGwYDVQQDExRNU0lUIEVudGVycHJpc2UgQ0EgMjAeFw0xNDA2MDMyMTQ2
NDVaFw0xNTA2MDMyMTQ2NDVaMAAwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGB
AK0VtILzXCuz4qNufhaRIzJ4SnibgSvyyZw5i3cyOpWxrKQ0NTk4xhAjW9QZMhev
Ic2DtjFlmm7HhELQXmn/vmccUKH+Gj/ZKoxh5iQRZOUBkarPsRWVdbojFHsCedbz
uDTR90qHM63VhNhOZ5a1mGJg8r4EbUXuOMgh+gfFYP/bAgMBAAGjggJ/MIICezA9
BgkrBgEEAYI3FQcEMDAuBiYrBgEEAYI3FQiDz4lNrfIChaGfDIL6yn2B4ft0gU+C
pMQ2g8+eawIBZAIBBDATBgNVHSUEDDAKBggrBgEFBQcDAjALBgNVHQ8EBAMCBaAw
GwYJKwYBBAGCNxUKBA4wDDAKBggrBgEFBQcDAjAuBgNVHREBAf8EJDAigiBkcm9p
ZC5yZWRtb25kLmNvcnAubWljcm9zb2Z0LmNvbTAdBgNVHQ4EFgQUVwJTpT7GXdAU
pxCsSn6E6t9rdfcwHwYDVR0jBBgwFoAU+N8R9KB9Uwu81Lf6HKRQqYvOlyUwgeIG
A1UdHwSB2jCB1zCB1KCB0aCBzoY0aHR0cDovL2NvcnBwa2kvY3JsL01TSVQlMjBF
bnRlcnByaXNlJTIwQ0ElMjAyKDMpLmNybIZLaHR0cDovL21zY3JsLm1pY3Jvc29m
dC5jb20vcGtpL21zY29ycC9jcmwvTVNJVCUyMEVudGVycHJpc2UlMjBDQSUyMDIo
MykuY3JshklodHRwOi8vY3JsLm1pY3Jvc29mdC5jb20vcGtpL21zY29ycC9jcmwv
TVNJVCUyMEVudGVycHJpc2UlMjBDQSUyMDIoMykuY3JsMIGlBggrBgEFBQcBAQSB
mDCBlTBABggrBgEFBQcwAoY0aHR0cDovL2NvcnBwa2kvYWlhL01TSVQlMjBFbnRl
cnByaXNlJTIwQ0ElMjAyKDQpLmNydDBRBggrBgEFBQcwAoZFaHR0cDovL3d3dy5t
aWNyb3NvZnQuY29tL3BraS9tc2NvcnAvTVNJVCUyMEVudGVycHJpc2UlMjBDQSUy
MDIoNCkuY3J0MA0GCSqGSIb3DQEBBQUAA4IBAQAiY6S6Tmo6ZGbn44VxHGzDzXWM
2qhUS2ZxRlxFoGSkXYvnYMThsgEaTPSfyJg2KE0GoWAj+dGGVWqoPdBh/thSVLz0
vc6L7LBYn22nM3cO5yRGUZjIQ0zqKm+Hdh6acayEqQz3nS0EcjOO0Lb3B+B/Yo9S
AIxFTAqqGjtEmECw5Ylye4jSztUhXzTj1l8cembnZJ3kGhkqev/HJH6FO/Mrm7d2
P9LigZ4q8mshX4Df3p0rdOMXalTPcKpz3Ge+TjBKBbB31rEzw9W5BkkhgWvU0eLJ
zT95x/AC1Q4df1f2b+rzhlhhE92vVzMUIvA/Ymv3YAhAFvHUILDI1Ww6nim6
              </EncodedCertificate>
          </System>
        </CA>
      </CertificateStore>

<!-- Wifi CSP -->
      <WiFi>
        <Profile name="Wifi_MSFTOPEN">
                <WlanXml>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MSFTOPEN&
lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MSFTOPEN&lt;/name&gt;&lt;/SSID&gt;&lt;/S
SIDConfig&gt;&lt;connectionType&gt;ESS&lt;/connectionType&gt;&lt;connectionMode&gt;auto&lt;/co
nnectionMode&gt;&lt;MSM&gt;&lt;security&gt;&lt;authEncryption&gt;&lt;authentication&gt;open&lt
;/authentication&gt;&lt;encryption&gt;none&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;/secur
ity&gt;&lt;/MSM&gt;&lt;/WLANProfile&gt;</WlanXml>
        </Profile>
```

```
      </WiFi>

<!-- OOBE Settings -->
      <Oobe>
         <AcceptTermsOfUse>true</AcceptTermsOfUse>
         <SkipSettings>true</SkipSettings>
         <SkipOnlineConsumerRegistration>true</SkipOnlineConsumerRegistration>
      </Oobe>

<!-- Time Zone Settings for India-->
      <AutomaticTime>
       <TimeZonePriority1>0x6B8</TimeZonePriority1>
      </AutomaticTime>

<!-- System Time CSP -->
      <MCSF>
        <AutomaticTime>
          <NTPRegularSyncInterval>1</NTPRegularSyncInterval>
          <NTPServers>time.windows.com&#xF000;time.nist.gov&#xF000;&#xF000;</NTPServers>
        </AutomaticTime>
      </MCSF>

<!-- Language Settings -->
      <BootUILanguage>en-us</BootUILanguage>


<!-- MDM Settings
      <EnterpriseExt>
        <MDM>
          <Server>p.manage-beta.microsoft.com</Server>
       </MDM>
      </EnterpriseExt>
-->
    </Common>
  </Settings>
 </Customization>
```

Certificate configuration (Updated in Windows Phone 8.1)

Windows Phone supports root, CA, and client certificate to be configured via MDM. CertificateStore configuration service provider is used to directly add/delete/query root and CA certificates, configure the device to enroll client certificate with certificate enrollment server that supports Simple Certificate Enrollment Protocol (SCEP). SCEP enrolled client certificates are used by Wi-Fi, VPN, email, and browser for certificate based client authentication. Each application has its own cert search criteria to allocate proper client cert for application usage. Additionally, the MDM enrollment client support enrolling the enterprise client certificate that contains the public key via the CertificateStore CSP with MDM enrollment server. The client certificate enrolled during MDM enrollment process could be used by native MDM client and native app download agent to do certificate based client authentication to MDM server and corporate server that host enterprise applications.

MDM server could also query and delete SCEP enrolled client certificate, or trigger a new enrollment request before the current certificate is expired. Refer Client Certificate Enrollment via SCEP section for more details.

Additionally, S/MIME signing certificate could also be enrolled via SCEP protocol.

Lastly, for organization that has higher security request, Windows Phone's virtual smart card (VSC) APIs will allow 3rd party to build an application to do VSC certificate provisioning and management.

Note that the CertificateStore CSP also accepts OMA CP (WAP) provisioning XML as used in enrollment provisioning. For a WAP provisioning XML sample, see the Response section of the Certificate enrollment policy web service section, earlier in this document. For more information about the configuration service provider, see CertificateStore configuration service provider, later in this document.

Enroll Client Certificate via Simple Certificate Enrollment Protocol

Windows Phone supports auto installing client certificates to enable Wi-Fi/VPN/Email/Browser certificate based authentication needs via Simple Certificate Enrollment Protocol (SCEP). This method could also be used to enroll S/MIME signing certificate.

NOTE 1: the SCEP enrolled certificate isn't protected by PIN. If you need PIN protected certificate, use virtual smart card certificate provision.

NOTE 2: MDM server could configure the device to store the certificate private key to Trusted Platform Module (TPM) to further protect the private key.

NOTE 3: The SCEP enrolled client certificate cannot be used for access secure website for application downloads. Instead, MDM enrolled client certificate could be used for access secure website for application downloads.

NOTE 4: the server should enroll SCEP client certificate first before sending other configuration to prevent configuring the device with no working profiles. E.g. only after certificate enroll succeeds (for example, polling enroll status value from ./Vendor/MSFT/CertificateStore/My/SCEP/<unique id>/Status via DM session), the server pushes down Wi-Fi/VPN/Email settings that requires client certificate.

NOTE 5: SCEP certificates with keylengths of 4096 is not supported.

To start with, MDM server will configure the device to enroll a specific certificate with SCEP server via CertificateStore CSP. The device will then initiate the certificate enrollment request.

The following steps shows the high level work flow of enrolling certificate via SCEP.

Certificate Enrollment precondition

Before MDM server sends the cert enroll request to the device, following tasks should be done.

1. IT admin has MDM server and CA Services setup
2. IT admin has hooked up enterprise CA with the SCEP server
3. MDM is configured to connect to SCEP server
4. Certificate templates are already stored in CA
5. SCEP server is configured with Certificate Templates for each Key Usage (Decipherment, Signature, Both)
6. The end user's phone is enrolled to be managed by MDM server
7. SCEP server must use the same CA cert for signing SCEP client cert and SCEP RA cert. And CA cert thumbprint should be provisioned as part of SCEP parameter configuration during MDM session.

Certificate Enrollment Steps

1. The MDM server generates the initial cert enroll DM request including challenge password, SCEP server URL, and other enrollment related parameters.
2. The policy is converted to the OMA DM request and sent to the device via the exposed CertificateStore CSP interface through DM session.

3. Trusted CA certificate should be installed directly during MDM request via CertificateStore CSP
4. At the device side, CertificateStore CSP accepts cert enroll request, invokes SCEP cert enroll client with parameters it received from the server. Note the actual enroll process is an asynchronous process from MDM enroll request.
5. The devices generates private/public key pair and SCEP request payload.
6. The device connects to Internet facing point exposed by SCEP server.
7. SCEP server creates the certificate that is signed with proper CA certificate and returns it to device.
   - The device supports the pending function to allow server side to do additional verification before issuing the cert. In this case, a pending status is sent back to the device. The device will periodically contact the SCEP server, based on preconfigured retry count and retry period parameters. Retrying ends when either:
     - A certificate is successfully received from the SCEP server
     - The SCEP server returns an error
     - The number of retries reaches the preconfigured limit
8. The cert is installed in the device. Browser, Wi-Fi, VPN, Email, and other first party applications have access to this certificate.
   - If MDM requested private key being stored in Trusted Process Module (TPM) (configured during enrollment request), the private key will be saved in TPM. Note that SCEP enrolled cert protected by TPM isn't guarded by a PIN.
9. If the certificate installed successfully, the device will trigger an OMA DM connection to the server to report the successful installation of client certificate via Generic Alert XML tag in the first DM package the device is sent to the server. The management server could leverage this information to decide next step of configuration, such as sending VPN/WiFi/Email configuration to the device. Note that sending success SCEP client certificate installation notification to the OMA server is a best effort action. No additional retry is built in for guaranteed delivery. The server could query SCEP status node in CertificateStore CSP to find out installation result as well.
   The following SyncML sample shows the message the device will send to the DM server when SCEP cert is installed successfully.

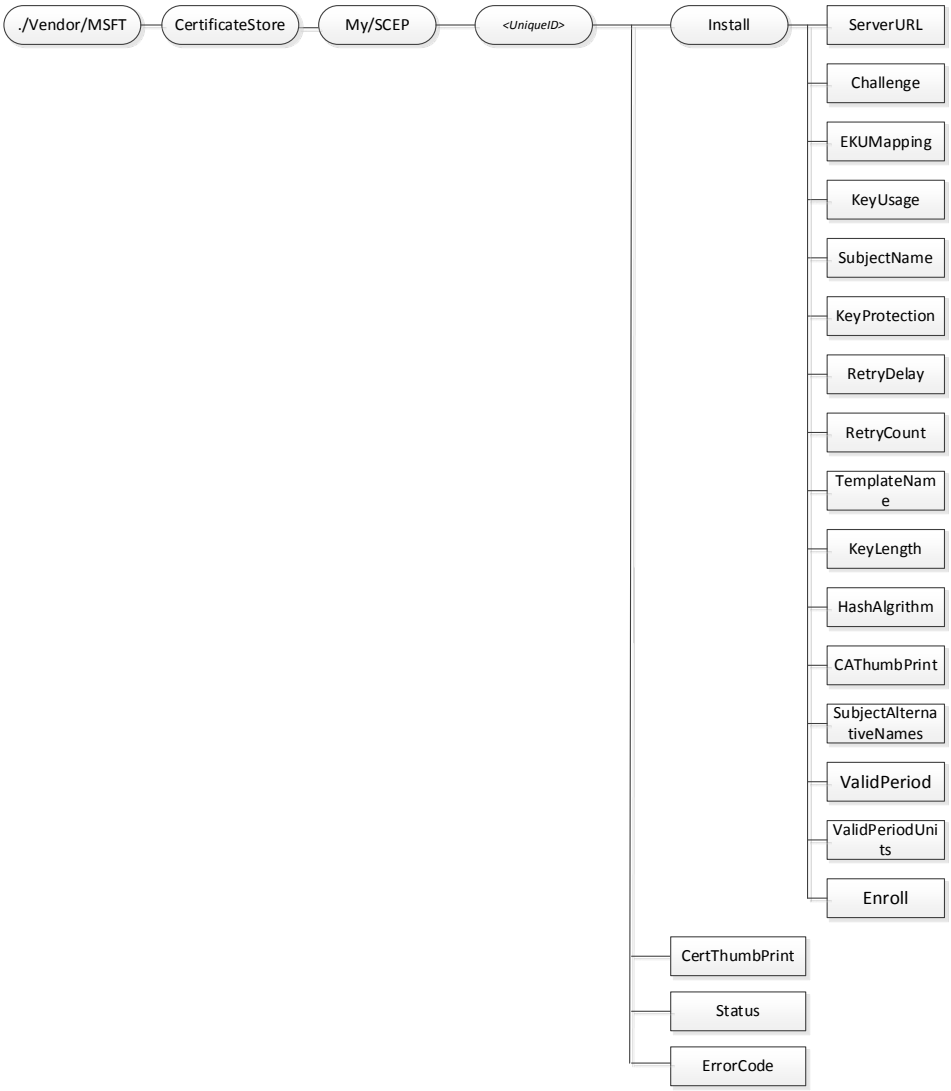
```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>1</MsgID>
      <Target>
         <LocURI>{unique device ID}</LocURI>
      </Target>
      <Source>
         <LocURI>https://www.thephone-company.com/mgmt-server</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Alert>
         <CmdID>1</CmdID>
         <Data>1201</Data> <!--client initiated session -->
       </Alert>
      <!-- Genric Alert for SCEP cert install result -->
       <Alert>
         <CmdID>1</CmdID>
```

```
         <Data>1226</Data> <!-- generic alert -->
         <Item>
           <Source>
              <LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/<unique
id>/Install/Enroll</LocURI>
           </Source>
           <Meta>
              <Type xmlns=”syncml:metinfo”>
com.microsoft:mdm.SCEPCertinstall.result</Type>
              <Format xmlns= “syncml:metinfo”>int</Format>
           </Meta>
           <Data>1</Data>
       </Alert>

      <!-- Basic DevInfo CSP information, truncated for simplicity -->
      <Replace>
         <CmdID>2</CmdID>
         <Item>
            <Source>
               <LocURI>./Devinfo/Man</LocURI>
            </Source>
            <Data>Device Factory</Data>
         </Item>
      </Replace>

      <Final />
   </SyncBody>
```

</SyncML>

CertifiateStore CSP Update to support SCEP certificate enrollment configuration

CertificateStore CSP is updated to accepting SCEP certificate enrollment related configuration. The following diagram shows SCEP corresponding parameters in CSP. Refer CertificateStore configuration service provider for detailed description for each node.



In summary, the MDM server could configure the device with SCEP enrollment server URL, SCEP challenge, pending retry schedule, key usage, key length, subject name, template name, hash algorithm, trusted CA certificate's thumb print, validation period, where private key should be saved, etc.

Exec DM command to Enroll node triggers the device to start the enrollment process with SCEP server.

Readonly CertThumbPrint node provide enrolled certificate's certificate thumbprint that is associated with the specific parent <unique id>.

Readonly Status node provides information on whether enroll succeeds, pending, or fails.

If the server needs to enroll multiple client certificates, it should add multiple <unique id> nodes which has different ID.

NOTE: The CertThumbPrint should be retrieved only after the status code = success

3. Update client certificate before SCEP enrolled certificate expires

For SCEP enrolled certificates, before they are expired, a new certificate with same cert property but new expiration date should be enrolled in the device ("reneweded"). This is achieved by MDM server sending a new challenge and Exec command to the to-be-expired cert path. Partially, the server should send following command to the device before the certificate indexed by <provider id 1> is expired:

- Send **Add** cmd to set a new challenge on ./Vendor/MSFT/CertificateStore/My/SCEP/<unique id 1>/Install/Challenge node as the previous challenge is removed shortly after first Exec command is accepted by the device.
- Send **Exec** cmd on ./Vendor/MSFT/CertificateStore/My/SCEP/<unique id 1>/Install/Enroll node

Add and Exec command should be wrapped in the same DM message. After the new certificate is installed in the device, the old certificate will be deleted by the device and query on the ./Vendor/MSFT/CertificateStore/My/SCEP/<unique id 1>/CertThumbPrint node will return new certificate's thumbprint.

NOTE: To make sure the "Exec" command is properly accepted and process at the device side in this case, the MDM server should check ./Vendor/MSFT/CertificateStore/My/SCEP/<unique id 1>/Status node value indicating the finishment of previous enrollment (succeed or fail) before sending Exec command again, e.g. the updating of certificate should start only after the previous certificate enrollment is done.

Inventory and Delete SCEP enrolled certificates

MDM server could query and delete SCEP enrolled certificate via CertificateStore CSP.

- Send **Get** cmd to ./Vendor/MSFT/CertificateStore/My/User node. The device will return the list of SCEP and/or user installed client certificates' thumbprint (hash). Send Get cmd to ./Vendor/MSFT/CertificateStore/My/User/<Certhash> node will return more detailed information for that certificate. For more details, refer CertificateStore configuration service provider
- Send **Delete** cmd to ./Vendor/MSFT/CertificateStore/My/SCEP/<unique id> node. The device will delete the certificate that has thumbprint = <unique id>

Enroll and manage MDM DM client certificate

As described in Connecting to the management infrastructure (enrollment) section, a MDM client certificate is enrolled via WSTEP protocol during MDM enrollment. The WSTEP enrolled certificate could be used by device as client certificate when do client cert based authentication to MDM server and to enterprise app content server for downloading LOB applications. The certificate needs to be renewed before it is expired.

In Windows Phone 8, the renew was done manually by user entering a valid corporate password at setting control panel's company account detail page.

In Windows Phone 8.1, an automatic certificate renew based on existing not expired client certificate is supported. To support such automatic certificate renew, enrollment server needs to be updated to support ROBO (renew on behalf of) – part of WSTEP protocol. The MDM server needs to be updated to be

able to send a ROBOSupport flag to the device during DM session to notify the device to use automatic certificate renew instead of manual renew via CertificateStore CSP.

NOTE: this ROBO based renew is only for certificate that is enrolled during MDM enrollment phase. For user manually installed certificate, no renew is built in. For SCEP enrolled certificate, refer Client Certificate Enrollment via Simple Certificate Enrollment Protocol (SCEP) section on how to provide updated certificate before current one is expired.

[More detail for ROBO renew will be provided later]

User manually install certificates

The end user could install a certificate via certificate file through email attachment or downloaded from browser. Refer http://www.microsoft.com/en-us/download/details.aspx?id=39262 to find out details on how the user could manually install a certificate. The certificate installer in the device handles the actual installation. It supports .cer, .p7b, .pem, and .pfx files.

Usage of user installed certificates

In Windows Phone 8, the user could install Root, CA, and client authentication certificates. In Windows Phone 8.1, the user can manually install Root, CA, client authentication certificates, and S/MIME encryption certificates (To be done in M3).

- The root and CA certificates are installed in the such way that any application could leverage it.
- The client certificate could be used by application that has the shared certificate (name to be updated) security capability.
- The encryption certificate is used by S/MIME client to encrypt outgoing message. Refer S/MIME Secure email doc for detailed description on S/MIME support.

Management of user installed certificate

MDM server could inventory and delete user installed certificate via CertificateStore CSP except S/MIME encryption certificate which isn't visible to MDM server.

- User installed root certificates are quer-iable/delete-able via ./Vendor/MSFT/CertificateStore/Root/System path
- User installed CA certificates are quer-iable/delete-able via ./Vendor/MSFT/CertificateStore/CA/System path
- User installed client certificates are quer-iable/delete-able via ./Vendor/MSFT/CertificateStore/User/My

  NOTE 1: There is no device setting control panel to view installed certificates.
  NOTE 2: There is no build certificate renew process for user installed certificates. The user needs to install an updated certificate before the current is expired to enable un-interrupt usage of certificates.

Company policy to disallow user manually install Root and CA certificates

MDM server could send policy (via OMA node ./Vendor/MSFT/PolicyManager/My/Security

/AllowManualRootCertificateInstallation) to disallow the user to install root or CA certificates. Refer PolicyManager configuration service provider for detailed description for each policy. If this policy is applied to the device, the user downloaded certificate file that contains root or CA certificate will not be installed in the device event if the certificate file also contains non root/CA certificates.

Virtual Smartcard Certificate Provisioning

For an organization that has more strict security requirement, such as two factor authentication with PIN protected TPM certificate, Windows Phone provides a set of virtual Smart Card (VSC) certificate APIs to allow 3rd party application to build a vSC certificate provision and management solution directly. Such certificate could be used by browser and S/MIME for client authentication and/or securing email.

Note: vSC certificates aren't managed by MDM server but 3rd party vSC application and vSC certificate provisioning server.

Global Certificate Revocation support

Windows Phone 8.1 has same certificate revocation support as Windows Phone 8. It supports both certificate revocation list (CRL) check and Online Certificate Status Protocol (OCSP).

Phone configuration

To get basic information about configuration settings of the phone, the enterprise management server can use two configuration service providers, which have been extended for Windows Phone. For more information, see DevDetail configuration service provider and DevInfo configuration service provider later in this document.

Wi-Fi configuration Windows Phone 8.1

The Wi-Fi configuration is supported in Windows Phone 8.1. The WiFi configuration service provider (CSP) provides functionality to add or delete Wi-Fi networks on a Windows Phone device. The CSP accepts a SyncML XML input and converts it to a network profile that is installed on the device. This profile enables the phone to connect to the Wi-Fi network when it is in range for Open, WEP, WPA2, PEAP-MSCHAPv2, and EAP-TLS/TTLS/SIM/AKA. Some Wi-Fi policies are configurable by MDM using PolicyManager configuration service provider.

VPN configuration Windows Phone 8.1

VPN is supported in Windows Phone 8.1. MDM servers can configure VPN profile via VPN configuration service provider. A few VPN policies are configurable by MDM using PolicyManager configuration service provider. MDM server could also configure IE intranet zone settings for VPN single sign on feature via registry key.

Email configuration

An Exchange Outlook account can be configured by using the ActiveSync configuration service provider. Other Internet email accounts can be configured by using the EMAIL2 configuration service provider.

Exchange Outlook account configuration

Note that when creating an Exchange Outlook account, the commands for individual nodes should be enclosed in an Atomic command. For more information and samples, see ActiveSync configuration service provider, later in this document.

Note: In Windows Phone 8.1 the AccountName is not properly set. To set the email account name, use ContentTypes/<GUID>/Name. This issue has been fixed for Windows Phone 8.1 GDR1 release.

Internet email account configuration

You can use the EMAIL2 configuration service provider to configure Internet POP3 and IMAP4 email accounts.

Note that you can also use this configuration service provider to enable Secure Sockets Layer (SSL) for incoming and outgoing email servers. To do so, use an unnamed tag: <parm name="8128000B" value="1"> for the incoming server, and <parm name="812C000B" value="1"> for the outgoing server.

Inventory cache handling

Some policies (such as device lock) configured by the enterprise management server can be changed by other entities, such as Exchange servers. The enterprise needs a way to ensure that the phone complies with company policies at all times. Instead of frequently querying all enterprise policy values to check for changes—which may cause battery drain and waste cellular bandwidth—Windows Phone 8 provides a tracking mechanism on the phone to allow the server to easily discover what has changed since its last query.

The server manages the client's cache to be in sync with the server side cache by using the NodeCache configuration service provider. For more information and examples, see NodeCache configuration service provider later in this document.

Coexistence of Exchange servers and enterprise management server

Windows Phone can be configured with one or more Exchange servers and one enterprise management server. The Exchange server(s) and enterprise management server could push down device lock policies and send remote wipe command to the phone. To make sure the phone is maintained as secured via those polices, the phone applies "most secure wins" logic: if different policy values are set by various servers, the phone ensures that the most secure value is applied. This design prevents a server from loosening the secure policies set by the other server. It also prevents a malicious sever from altering legitimate company server policies. The client also ensures that if one server account is removed from the phone, the next most secure policy value that is set by other servers is applied.

All this logic is built-in at the client side. The server only needs to push down whatever policy values are set by IT administration. The phone will ensure that either the exact value or a more secure value is enforced on the phone. The enterprise management server can also use the PolicyManager CSP to query either actual policy values applied to the phone or policy values pushed down by the server. Notice that this query function is only available to enterprise management server, not via Exchange.

If users want to fully disassociate from the enterprise, they should do both of the following:

- Remove the workplace account, which removes management sever applied company policies/settings and installed enterprise applications and associated enterprise token.
- Delete the company Outlook Exchange account (or other corporate email account) in order to delete corporate email and policies set by Exchange if the account isn't created by MDM server.

In addition to device lock polices, Windows Phone also supports other Exchange ActiveSync polices. The following table shows the summary of polices supported via a dedicated management server and Exchange.

Policy	Exchange	Enterprise Mgmt Server
Password (device lock) required	Yes	Yes
Simple password	Yes	Yes
Alphanumeric password	Yes	Yes
Minimum password length	Yes	Yes
Minimum password complex characters	Yes	Yes
Password expiration	Yes	Yes
Password history	Yes	Yes
Maximum password attempts before wipe	Yes	Yes
Device inactivity time-out	Yes	Yes
IRM enabled (Exchange server-side policy)	Yes	No
Remote wipe full device	Yes	Yes
Require device encryption (new in Windows Phone 8)	Yes	Yes
Disable storage card (new in Windows Phone 8)	Yes	Yes
Remote update of LOB apps (new in Windows Phone 8)	n/a	Yes
Remote or local delete of MDM association (removes all LOB apps/data & MDM server applied enterprise policies and configuration) (updated in Windows Phone Windows Phone 8.1)	n/a	Yes
Allow developer unlock (new in Windows Phone 8.1)	n/a	Yes
Allow using Microsoft account for non-email related connection authentication and services (new in Windows Phone 8.1)	n/a	Yes
Allow adding non Microsoft Accounts manually (new in Windows Phone 8.1)	n/a	Yes
Allow app store (new in Windows Phone 8.1)	n/a	Yes

Policy	Exchange	Enterprise Mgmt Server
Specify application restrictions (new in Windows Phone 8.1)	n/a	Yes
Allow NFC (new in Windows Phone 8.1)	n/a	Yes
Allow manual root certificate installation (new in Windows Phone 8.1)	n/a	Yes
Allow Wi-Fi offloading (new in Windows Phone 8.1)	n/a	Yes
Allow Wi-Fi hotspots reporting (new in Windows Phone 8.1)	n/a	Yes
Allow manual Wi-Fi configuration (new in Windows Phone 8.1)	n/a	Yes
Allow Telemetry (new in Windows Phone 8.1)	n/a	Yes
Allow Wi-Fi hotspots reporting (new in Windows Phone 8.1)	n/a	Yes
Allow Copy and Paste (new in Windows Phone 8.1)	n/a	Yes
Allow Bluetooth (new in Windows Phone 8.1)	Yes	Yes
Allow Internet sharing (new in Windows Phone 8.1)	Yes	Yes
Allow Camera (new in Windows Phone 8.1)	Yes	Yes
Allow Data Protection under PIN Lock (new in Windows Phone 8.1 GDR2)	No	Yes
Enable Fully-managed VPN Settings on Devices (new in Windows Phone 8.1 GDR2)	No	Yes
Allow Enterprise Over-ride of Anti-theft Mode	No	Yes
Allow customization of Cellular App Download Limit	No	Yes
Allow customization of Wi-Fi Scan Frequency	No	Yes
Disable Task Switcher Control on Devices	No	Yes

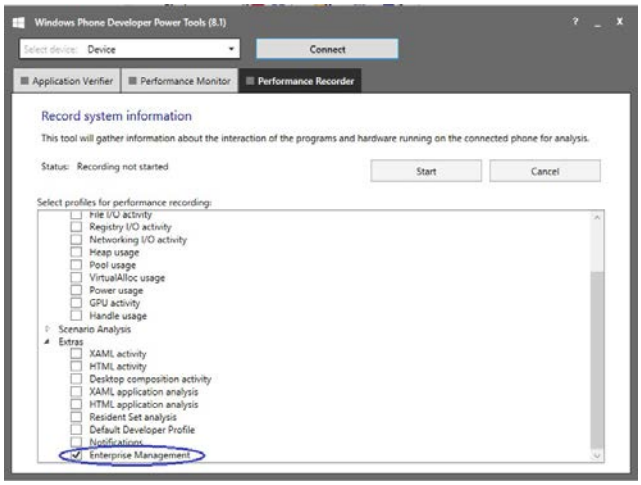
Logging support for Enterprise server creation (New in Windows Phone 8.1)

MDM logging is enabled in Windows Phone 8.1 in order to help MDM ISV self-debugging issues during development cycle. The log file focuses on logging transaction errors during MDM enrollment, DM session, and SCEP certificate enrollment. The corresponding ETW logs is exposed via Windows Phone Developer Power Tools (8.1) in Windows Phone 8.1 SDK.

Retrieve MDM logs

The following steps describe how to use Windows Phone 8.1 SDK Power Tools to get MDM logs.

1) Install Windows Phone 8.1 SDK, launch Power tools.
2) Under Performance Recorder, check Enterprise Management under Extra category.
3) Select a device (emulator image or dev unlocked retail device connected to PC),
4) Tap Start button to start log.
5) Run enterprise scenarios.
6) Tap Stop button to save ETW log to a local location.



View ETW logs

Use Windows Performance Analyzer to view the log.

The logging information is saved in ETL file. The MDM developer should use the Win8.1 Windows® Performance Analyzer (WPA) to view the log. Windows 7 or 8 machine is required to use this tool.

WPA is part of Windows Performance Toolkit . Included in the Windows® Assessment and Deployment Kit (Windows ADK) and the Windows Software Development Kit (SDK). Windows® Performance Analyzer

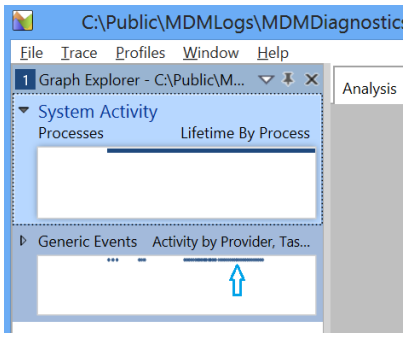
(WPA) is a tool that creates graphs and data tables of Event Tracing for Windows (ETW) events that are recorded by Windows® Performance Recorder (WPR), Xperf, or an assessment that is run in the Assessment Platform.

MSDN reference link for WPA: http://msdn.microsoft.com/en-us/library/windows/desktop/hh448170.aspx

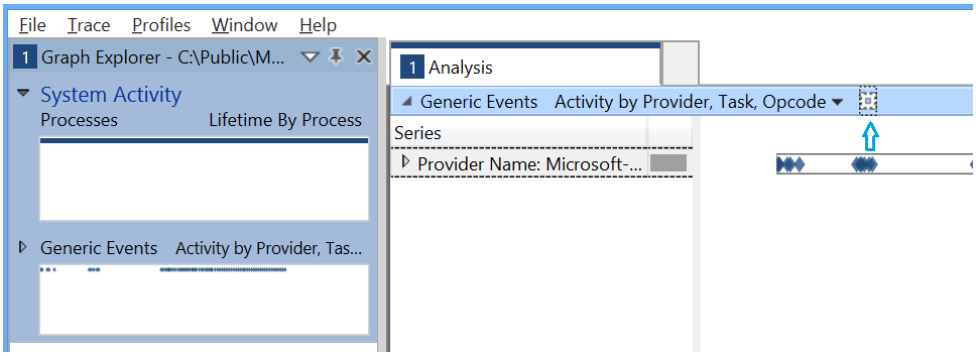
The 8.1 version of WPA tool that support all fields we need ISV to view MDM logs. It can be downloaded here: http://www.microsoft.com/en-us/download/details.aspx?id=39982

Steps to use WPA tool to view MDM log file

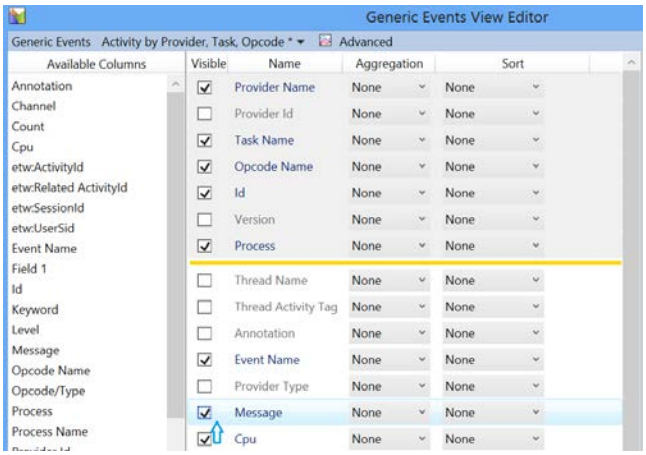
1. Use Windows Performance Analyzer to open the etl file.
2. In WPA's "Graph Explorer" window, expand "System Activity". A "Generic Events" sub window will be displayed.



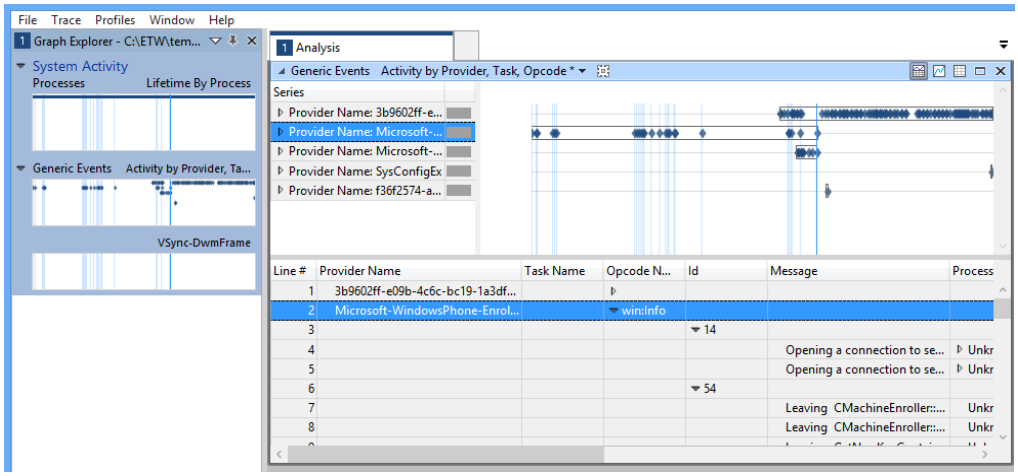
3. Double click the graphic bars in "Generic Events" window. An "Analysis" window will show up at right side



4. In "Analysis" window, click "Open View Editor" Icon. A "Generic Events View Editor" window will pop up.



5. In "Generic Events View Editor" window, make sure "Message" check box is checked. Click "Apply" button.
6. "message" field in "Analysis" window provides MDM specific log message under various providers. You could copy/paste information in the sheet to other file for further analysis.



ETW logs for MDM enrollment, MDM client cert renew process

- Microsoft-WindowsPhone-Enrollment-API-Provider

SCEP certificate enrollment

- Microsoft-WindowsPhone-SCEP-Provider

VPN Configuration

- Microsoft-WindowsPhone-CmCspVpnPlus

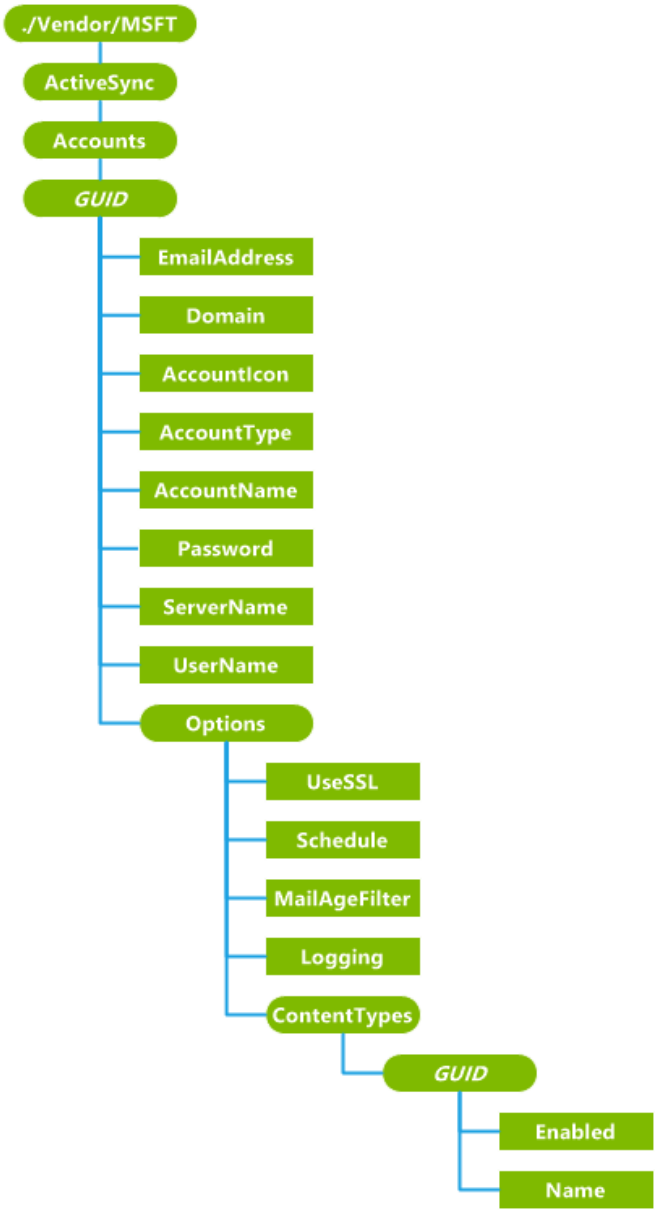
Configuration service provider reference

The following terms are used to describe the configuration service provider functionalities:

- **Required** – If a parameter is marked required, it means the node must exist to ensure that configuration succeeds.
- **Optional** – If a parameter is marked Optional, it means the configuration will succeed with or without that value. In other words, the value may be required to be set at some point, for proper operation, but the phone won't throw an error if the value isn't included in the provisioning request payload. Be sure to check the description of each node to understand whether an "optional" node is required in certain scenarios.
- **Get** – A device management command to query the phone settings.
- **Add** – A device management command to Add a new setting to the phone.
- **Replace** – A device management command to Update the phone value to a new value.
- **Delete** –A device management command to delete a phone setting.

ActiveSync configuration service provider

The following image shows the ActiveSync configuration service provider management object.



./Vendor/MSFT/ActiveSync

The root node for ActiveSync CSP. Supported operation: Get.

Accounts

The root node for all ActiveSync accounts. Supported operations: Get.

<GUID>

Defines a specific ActiveSync account. A globally unique identifier (GUID) must be generated for each ActiveSync account on the phone. Supported operations: Get, Add, and Delete.

When managing over OMA DM, make sure to always use a unique GUID. When an account is deleted, creating a new account with the same email is a different account from the previous one. Different accounts should have different GUIDs.

Braces { } are required around the GUID in the EMAIL2 configuration service provider. In OMA Client Provisioning, you can type the braces. For example:

```
<characteristic type="{C556E16F-56C4-4EDB-9C64-D9469EE1FBE0}"/>
```

For OMA DM, you must use the ASCII values of %7B and %7D for the opening and closing braces, respectively. For example, if the GUID is "C556E16F-56C4-4EDB-9C64-D9469EE1FBE0", you would type:

```
<Target>
   <LocURI>
      ./Vendor/MSFT/EMAIL2/%7BC556E16F-56C4-4EDB-9C64-D9469EE1FBE0%7D
   </LocURI>
</Target>.
```

EmailAddress

Required. A character string that specifies the email address associated with the Exchange ActiveSync account. Supported operations: Get, Replace, Add (cannot Add after the account is created).

This email address is entered by the user during setup and must be in the fully qualified email address format, for example, "someone@example.com".

Domain

Optional for Exchange. Specifies the domain name of the Exchange server. Supported operations: Get, Replace, Add, and Delete.

AccountIcon

Required. A character string that specifies the location of the icon associated with the account. Supported operations: Get, Replace, Add (cannot Add after the account is created).

The account icon can be used as a tile in the Start list or an icon in the applications list under Settings > Email + accounts. Some icons are already provided on the phone. The suggested icon for POP/IMAP or generic ActiveSync accounts is at res://AccountSettingsSharedRes{*ScreenResolution*}!%s.genericmail.png.
The suggested icon for Exchange Accounts is at res://AccountSettingsSharedRes{*ScreenResolution*}!%s.office.outlook.png. Custom icons can be added if desired.

AccountTypes

Required. A character string that specifies the account type. Supported operations: Get, Add (cannot Add after the account is created).

This value is entered during setup and cannot be modified once entered. An Exchange account is indicated by the string value "Exchange".

AccountName

Required. A character string that specifies the name that refers to the account on the phone. Supported operations: Get, Replace, Add (cannot Add after the account is created).

Note: In Windows Phone 8.1 the AccountName is not properly set. To set the email account name, use ContentTypes/<GUID>/Name. This issue has been fixed for Windows Phone 8.1 GDR1 release.

Password

Required. A character string that specifies the password for the account. Supported operations: Get, Replace, Add, and Delete. For the Get command, only asterisks are returned.

ServerName

Required. A character string that specifies the server name used by the account. Supported operations: Get, Replace, Add (cannot Add after the account is created).

UserName

Required. A character string that specifies the user name for the account. Supported operations: Get, Add (cannot Add after the account is created).

The user name cannot be changed after a sync has been successfully performed. The user name can be in the fully qualified format "someone@example.com", or just "username", depending on the type of account created. For most Exchange accounts, the user name format is just "username", whereas for Microsoft, Google, Yahoo, and most POP/IMAP accounts, the user name format is "someone@example.com".

UseSSL

Optional. A character string that specifies whether SSL is used. The default is "1" (used). Supported operations: Get, Replace, Add (cannot Add after the account is created).

The value of "0" specifies that SSL is not used. The default value of "1" specifies that SSL is used.

Schedule

Required. A character string that specifies the time until the next sync is performed, in minutes. Supported operations: Get, Replace.

The default value of "−1" specifies that a sync will occur as items are received. Other valid values are:

- 0 specifies that all syncs must be performed manually.
- 15 – sync every 15 minutes
- 30 – sync every 30 minutes
- 60 – sync every 60 minutes

MailAgeFilter

Required. A character string that specifies the time window used for syncing email items to the phone. The default is 3. Supported operations: Get, Replace. The valid values are:

- 0 – No age filter is used, and all email items are synced to the phone.
- 2 – Only email up to three days old is synced to the phone.
- 3 – The default value. Email up to a week old is synced to the phone.
- 4 – Email up to two weeks old is synced to the phone.
- 5 – Email up to a month old is synced to the phone.

Logging

A character string that specifies whether diagnostic logging is enabled and at what level. The default is "0" (disabled). Supported operations: Get, Replace, Add (cannot Add after the account is created).

The default value of "0" specifies that logging is disabled (off). A value of "2" enables advanced logging. The only supported values are 0 and 2.

Logging is set to off by default. The user might be asked to set this to Advanced when having a sync issue that customer support is investigating.

ContentTypes/<GUID>

Defines the type of content to be individually enabled/disabled for sync. Supported operations: Get, Replace, Delete, Add.

The *GUID* values allowed are as follows:

- Email: "{c6d47067-6e92-480e-b0fc-4ba82182fac7}"
- Contacts: "{0dd8685c-e272-4fcb-9ecf-2ead7ea2497b}"
- Calendar: "{4a5d9fe0-f139-4a63-a5a4-4f31ceea02ad}"
- Task: {783ae4f6-4c12-4423-8270-66361260d4f1}

ContentTypes/<GUID>/Enabled

Required. A character string that specifies whether sync is enabled or disabled for the selected content type. The default is "1" (enabled). Supported operations: Get, Replace, Add (cannot Add after the account is created).

A value of "0" specifies that sync for email, contacts, calendar, or tasks is disabled. The default value of "1" specifies that sync is enabled.

ContentTypes/<GUID>/Name

Required. Specifies the name of the content type as a string. Supported operations: Get, Replace, Add (cannot Add after the account is created).

Example

The following sample shows how to configure Outlook ActiveSync account settings.

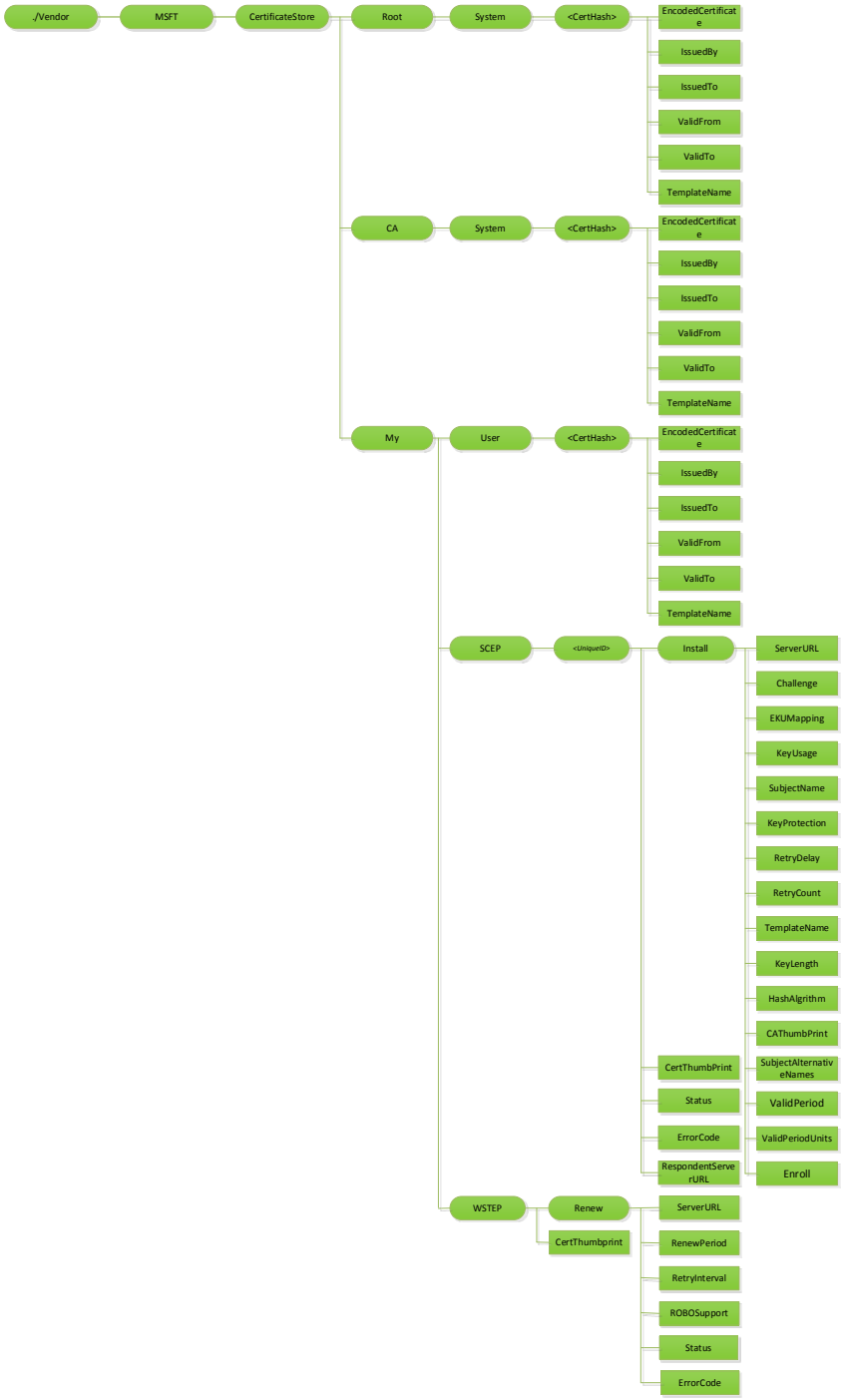
```
<Atomic>
   <CmdID>13</CmdID>
   <Add>
      <CmdID>4</CmdID>
      <Item>
         <Target>
            <LocURI>
           ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/Domain
            </LocURI>
         </Target>
         <Data>contoso</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>
      ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/AccountType
            </LocURI>
         </Target>
         <Data>Exchange</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>7</CmdID>
      <Item>
         <Target>
            <LocURI>
         ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/Password
            </LocURI>
         </Target>
         <Data>Password1</Data>
```

```
   </Item>
</Add>
<Add>
   <CmdID>6</CmdID>
   <Item>
      <Target>
         <LocURI>
   ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/AccountName
         </LocURI>
      </Target>
      <Data>TestAccount</Data>
   </Item>
</Add>
<Add>
   <CmdID>9</CmdID>
   <Item>
      <Target>
         <LocURI>
      ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/UserName
         </LocURI>
      </Target>
      <Data>user</Data>
  </Item>
</Add>
<Add>
   <CmdID>8</CmdID>
   <Item>
      <Target>
         <LocURI>
    ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/ServerName
         </LocURI>
      </Target>
      <Data>contoso.com</Data>
   </Item>
</Add>
<Add>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>
  ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/EmailAddress
         </LocURI>
      </Target>
      <Data>user@contoso.com</Data>
   </Item>
</Add>
<Add>
   <CmdID>10</CmdID>
   <Item>
      <Target>
         <LocURI>
./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/Options/UseSSL
         </LocURI>
      </Target>
      <Data>1</Data>
   </Item>
</Add>
<Replace>
   <CmdID>11</CmdID>
   <Item>
      <Target>
         <LocURI>
```

```
./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-77fb2b96c42d%7D/Options/Schedule
            </LocURI>
         </Target>
         <Data>15</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>12</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/ActiveSync/Accounts/%7Bac63781d-9bf2-4442-a7f2-
77fb2b96c42d%7D/Options/MailAgeFilter
            </LocURI>
         </Target>
         <Data>3</Data>
      </Item>
   </Replace>
</Atomic>
```

CertificateStore configuration service provider (Updated in Windows Phone 8.1)

The following diagram shows the CertificateStore configuration service provider management object in tree format.


./Vendor
MSFT
CertificateStore
Root
System
<CertHash>
EncodedCertificate
IssuedBy
IssuedTo
ValidFrom
ValidTo
TemplateName
CA
System
<CertHash>
EncodedCertificate
IssuedBy
IssuedTo
ValidFrom
ValidTo
TemplateName
My
User
<CertHash>
EncodedCertificate
IssuedBy
IssuedTo
ValidFrom
ValidTo
TemplateName
SCEP
<UniqueID>
Install
ServerURL
Challenge
EKUMapping
KeyUsage
SubjectName
KeyProtection
RetryDelay
RetryCount
TemplateName
KeyLength
HashAlgrithm
CAThumbPrint
SubjectAlternativeNames
ValidPeriod
ValidPeriodUnits
Enroll
CertThumbPrint
Status
ErrorCode
RespondentServerURL
WSTEP
Renew
ServerURL
RenewPeriod
RetryInterval
ROBOSupport
Status
ErrorCode
CertThumbprint

Root/System
Defines the certificate store that contains root, or self-signed, certificates. Supported operation is Get.

CA/System
Defines the certificate store that contains cryptographic information, including intermediary certification authorities. Supported operation is Get.

My/User
Defines the certificate store that contains public key for client certificate. This is only used by enterprise server to push down the public key of the client cert. The client cert is used by the phone to authenticate itself to the enterprise server for device management and enterprise app downloading. Supported operation is Get.

<CertHash>
Defines the SHA1 hash for the certificate. The 20-byte value of the SHA1 certificate hash is specified as a hexadecimal string value. Supported operations are Add, Delete, and Get.

<CertHash>/EncodedCertificate
Required. Specifies the X.509 certificate as a Base64-encoded string. Supported operation is Add**,** Get. The Base-64 string value cannot include extra formatting characters such as embedded linefeeds, etc.

<CertHash>/IssuedBy
Required. Returns the name of the certificate issuer. Supported operation is Get. This is equivalent to the *Issuer* member in the CERT_INFO data structure.

<CertHash>/IssuedTo
Required. Returns the name of the certificate subject. Supported operation is Get. This is equivalent to the *Subject* member in the CERT_INFO data structure.

<CertHash>/ValidFrom
Required. Returns the starting date of the certificate's validity. Supported operation is Get. This is equivalent to the *NotBefore* member in the CERT_INFO structure.

<CertHash>/ValidTo
Required. Returns the expiration date of the certificate. Supported operation is Get. This is equivalent to the *NotAfter* member in the CERT_INFO structure

<CertHash>/TemplateName
Required. Returns the certificate template name. Supported operation is Get.

My/SCEP
Required for SCEP certificate enrollment. The parent node grouping the SCEP cert related settings. Supported operation is Get.

My/SCEP/<UniqueID>
Required for SCEP certificate enrollment. A unique ID to differentiate different certificate enrollment requests. Format is node. Supported operations are Get, Add, Delete

My/SCEP/<UniqueID>/Install
Required for SCEP certificate enrollment. Parent node to group SCEP cert install related request. Format is node. Supported operation is Add, Delete.

NOTE: though the children nodes under Install support Replace commands, once the Exec command is sent to the device, the device will take the values which are set when the Exec command is accepted. The server should not expect the node value change after Exec command is accepted will

impact the current undergoing enrollment. The server should check the Status node value and make sure the device is not at unknown stage before changing children node values.

My/SCEP/<UniqueID>/Install/ServerURL
Required for SCEP certificate enrollment. Specify the cert enrollment server. The server could specify multiple server URLs separated by semicolon. Format is chr. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/Challenge
Required for SCEP certificate enrollment. B64 encoded SCEP enrollment challenge. Format is chr. Supported operations are Get, Add, Replace, Delete. Challenge will be deleted shortly after the Exec command is accepted.

My/SCEP/<UniqueID>/Install/RetryCount
Optional. Special to SCEP. Specify device retry times when the SCEP sever sends pending status. Format is int. Default value is 3. Max value: the value cannot be larger than 30. If it is larger than 30, the device will use 30.

The min value is 0 which means no retry. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/RetryDelay
Optional. When the SCEP server sends pending status, specify device retry waiting time in minutes. Default value is: 5

The min value is 1. . Format is int. Supported operations are Get, Add, Delete.

My/SCEP/<UniqueID>/Install/TemplateName
Optional. OID of certificate template name. Note that this name is typically ignored by the SCEP server, therefore the MDM server typically doesn't need to provide it. Format is chr. Supported operations are Get, Add, Delete.

My/SCEP/<UniqueID>/Install/KeyUsage
Required for enrollment. Specify the key usage bits (0x80, 0x20, 0xA0, etc.) for the certificate in decimal format. The value should at least have second (0x20) or forth (0x80) or both bits set. If the value doesn't have those bits set, configuration will fail. Format is int. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/KeyLength
Required for enrollment. Specify private key length (RSA). Format is int. Valid value: 1024, 2048, 4096. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/HashAlgorithm
Required for enrollment. Hash algorithm family (SHA-1, SHA-2, SHA-3) specified by MDM server. If multiple hash algorithm families are specified, they must be separated via +. Format is chr. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/CAThumbprint
Required. Specify root CA thumbprint. It is a 20-byte value of the SHA1 certificate hash specified as a hexadecimal string value. When client authenticates SCEP server, it checks CA cert from SCEP server whether match with this cert. If not match, fail the authentication. Format is chr. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/SubjectName
Required. Specify the subject name. Format is chr. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/SubjectAlternativeNames

Optional. Specify subject alternative name. Multiple alternative names could be specified by this node. Each name is the combination of name format+actual name. Refer name type definition in MSDN. Each pair is separated by semicolon. E.g. multiple SAN are presented in the format of <nameformat1>+<actual name1>;<name format 2>+<actual name2>. Format is chr. Supported operations are Get, Add, Delete, Replace.

NOTE: Windows Phone 8.1 only supports configuring the following AlternativeName types:

XCN_CERT_ALT_NAME_RFC822_NAME = 2
XCN_CERT_ALT_NAME_DNS_NAME = 3
XCN_CERT_ALT_NAME_URL = 7
XCN_CERT_ALT_NAME_REGISTERED_ID = 9
XCN_CERT_ALT_NAME_USER_PRINCIPLE_NAME = 11.

All other types are unsupported and will result in an error.

My/SCEP/<UniqueID>/Install/ValidPeriod

Optional. Specify the units for valid period. Valid values are: Days(Default), Months, Years. Format is chr. Supported operations are Get, Add, Delete, Replace.

NOTE: The device only sends the MDM server expected certificate validation period (ValidPeriodUnits + ValidPerio) the SCEP server as part of certificate enrollment request. It is the server's decision on how to use this valid period to create the certificate.

My/SCEP/<UniqueID>/Install/ValidPeriodUnits

Optional. Specify desired number of units used in validity period. Subjected to SCEP server configuration. Default is 0. The units are defined in ValidPeriod node. Note the valid period specified by MDM will overwrite the valid period specified in cert template. For example, if ValidPeriod is days and ValidPeriodUnits is 30, it means the total valid duration is 30 days. Format is int. Supported operations are Get, Add, Delete, Replace.

NOTE: The device only sends the MDM server expected certificate validation period (ValidPeriodUnits + ValidPerio) the SCEP server as part of certificate enrollment request. It is the server's decision on how to use this valid period to create the certificate.

My/SCEP/<UniqueID>/Install/EKUMapping

Required. Specify extended key usages. Subjected to SCEP server configuration. The list of OIDs are separated by plus "+". Sample format: OID1+OID2+OID3. Format is chr. Supported operations are Get, Add, Delete, Replace.

My/SCEP/<UniqueID>/Install/KeyProtection

Optional. Specify where to keep the private key. Note that even it is protected by TPM, it is not guarded with TPM PIN. SCEP enrolled cert doesn't support TPN PIN protection. Supported values: 1 – private key protected by phone TPM,

3 (default) – private key saved in OS (not protected by TPM), 2 – private key protected by phone TPM if the device supports TPM. All Windows Phone 8.1 devices support TPM and will treat value 2 as 1 Format is int. Supported operations are Get, Add, Delete, Replace

My/SCEP/<UniqueID>/Install/Enroll

Required. Trigger the device to start the cert enrollment. The MDM server could later query the device to find out whether new cert is added. Format is null, e.g. this node doesn't contain a value. Supported operation is Exec.

My/SCEP/<UniqueID>/Status

Required. Specify the latest status for the certificate due to enroll request. Format is chr. Supported operation is Get. Valid value:

1 – finished successful

2 – pending (the device hasn't finish the action but receives the SCEP server pending response)

32 – unknown

16 - action failed

My/SCEP/<UniqueID>/ErrorCode

Optional. The integer value that indicates the HRESULT of the last enrollment error code. Supported operation is Get.

My/SCEP/<UniqueID>/CertThumbprint

Optional. Specify the current cert's thumbprint if certificate enrollment succeeds. It is a 20-byte value of the SHA1 certificate hash specified as a hexadecimal string value. Format is chr. Supported operation is Get.

My/WSTEP

Required for MDM enrolled device. The parent node that hosts MDM enrollment enrolled client certificate related settings that is enrolled via WSTEP. The nodes under WSTEP are mostly for MDM client certificate renew request. Format is node. Supported operation is Get.

My/WSTEP/CertThumbprint

Optional. Return the current MDM client certificate's thumbprint. If renew succeeds, it shows renewed cert's thumbprint. If renew doesn't succeed or in progress, it shows the thumbprint of cert that needs to be renewed. Format is chr. Supported operation is Get.

My/WSTEP/Renew

Optional. Parent node to group renew related settings. Supported operation is Get.

My/WSTEP/Renew/ServerURL

Optional. Specify the cert renewal server URL. If this node doesn't exist, the client will use the initial certificate enrollment URL. Supported operations are Add, Get, Delete, Replace.

Note: The renewal process follows the same steps as device enrollment, which means that it starts with Discovery service, followed by Enrollment policy service, and then Enrollment web service.

My/WSTEP/Renew/ ROBOSupport

Optional. Notify the client whether MDM enrollment server supports ROBO auto certificate renew. The datatype for this node is bool.

For MDM enrolled with On-premise authentication method, by default, the device will use manual certificate renew. If the server sets this value to true, the device will use ROBO as renew method at background, no user action is needed.

For MDM enrolled with federated authentication, ROBO is the only supported renewal method. If the server sets this node value to be false or delete this node for federated enrolled device, the configuration will fail.

Supported operations are Add, Get, Delete, Replace.

NOTE: when set renew schedule over SyncML DM commands to ROBOSupport, RenewalPeriod, and RetryInterval, those command should be wrapped in Atomic command.

My/WSTEP/Renew/RenewalPeriod

Optional. The time (in days) before the MDM certificate is expired to trigger the client to initiate the MDM client certificate renew process. The MDM server could set and update the renew period. This parameter applies to both manual cert renewal and ROBO cert renewal. It is recommended that renew period should be set a couple months before cert expire to ensure the cert get successfully renewed with data connectivity.

Supported operations are Add, Get, Delete, Replace.

Default value is 42.

Datatype of this node value is int.

Valid value: 1 – 1000

NOTE: when set renew schedule over SyncML DM commands to ROBOSupport, RenewalPeriod, and RetryInterval, those command should be wrapped in Atomic command.

My/WSTEP/Renew/RetryInterval

Optional. This parameter specifies retry interval when previous renew failed (in days). It applies to both manual cert renewal and ROBO automatic cert renewal. Retry schedule will stop at cert expiration date.

For ROBO renew failure, the client will retry the renew periodically till the device reach cert expiration date. This parm specify the ROBO renew failure retry waiting period.

For manual retry failure, there is no built in renew failure retry, the user on the other side could retry later. At next scheduled cert renew retry time, the device will prompt credential to be expired soon dialog again. Supported operations are Add, Get, Delete, Replace.

Default value is 7.

Datatype of this node value is int.

Valid value: 1 – 1000

NOTE: when set renew schedule over SyncML DM commands to ROBOSupport, RenewalPeriod, and RetryInterval, those command should be wrapped in Atomic command.

My/WSTEP/Renew/Status

Required. Show the latest action status for this certificate.

Datatype of this node value is int.

Supported option is Get.

Supported value:

0 – not start

1 – renewal in progress

2 – renew succeeded

3 – renew failed

My/WSTEP/Renew/ErrorCode

Optional. If certificate renew fails, this integer value indicates the HRESULT of the last error code during renew process. Datatype of this node value is int.

Supported option is Get.

Examples

Adding a root certificate via the MDM server

```
<Add>
   <CmdID>1</CmdID>
   <Item>
      <Target>
         <LocURI>
./Vendor/MSFT/CertificateStore/Root/System/<CertificateHashInsertedhere>/EncodedCertificate
          </LocURI>
      </Target>
      <Data>B64EncodedCertInsertedHere</Data>
      <Meta>
         <Format xmlns="syncml:metinf">b64</Format>
      </Meta>
   </Item>
</Add>
```

Iterating all installed client certificates

```
<Get>
   <CmdID>1</CmdID>
   <Item>
      <Target>
         <LocURI>
               ./Vendor/MSFT/CertificateStore/My/User?list=StructData
          </LocURI>
      </Target>
   </Item>
</Get>
```

Deleting a root certificate

```
<Delete>
   <CmdID>1</CmdID>
   <Item>
      <Target>
         <LocURI>
               ./Vendor/MSFT/CertificateStore/Root/System/<CertificateHashInsertedHere>
          </LocURI>
      </Target>
   </Item>
</Delete>
```

Configuring the device to enroll a client certificate via SCEP

Please note SCEP certification enrollment configuration request DM commands should be wrapped within Atomic command to make sure enrollment execution isn't triggered till all settings are configured.

```
<Atomic>
<CmdID>100</CmdID>
<Add>
   <CmdID>1</CmdID>
   <Item>
        <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP1</LocURI></Target>
        <Meta>
              <Format xmlns="syncml:metinf">node</Format>
        </Meta>
   </Item>
</Add>
<Add>
    <CmdID>2</CmdID>
    <Item>
        <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/RetryCount</LocURI></Target>
    <Meta>
              <Format xmlns="syncml:metinf">int</Format>
    </Meta>
            <Data>1</Data>
    </Item>
</Add>
<Add>
         <CmdID>3</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/RetryDelay</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>1</Data>
         </Item>
</Add>
<Add>
         <CmdID>4</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/KeyUsage</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>160</Data>
         </Item>
</Add>
<Add>
         <CmdID>5</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/KeyLength</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>1024</Data>
         </Item>
</Add>
<Add>
         <CmdID>6</CmdID>
         <Item>
```

```
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/HashAlgorithm</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>SHA-1</Data>
         </Item>
</Add>
<Add>
         <CmdID>7</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/SubjectName</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>CN=AnnaLee</Data>
         </Item>
</Add>
<Add>
         <CmdID>8</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/SubjectAlternativeNames</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>11+tom@MyDomain.Contoso.com;3+MyDomain.Contoso.com</Data>
         </Item>
</Add>
<Add>
         <CmdID>9</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/ValidPeriod</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>Years</Data>
         </Item>
</Add>
<Add>
         <CmdID>10</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/ValidPeriodUnits</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>1</Data>
         </Item>
</Add>
<Add>
         <CmdID>11</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP
```

```
1/Install/EKUMapping</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>1.3.6.1.4.1.311.10.3.12+1.3.6.1.4.1.311.10.3.4+1.3.6.1.4.1.311.20.2.2</Data>
         </Item>
</Add>
<Add>
         <CmdID>12</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/KeyProtection</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>3</Data>
         </Item>
</Add>
<Add>
         <CmdID>13</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/ServerURL</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>https://contoso.com/certsrv/ctcep.dll</Data>
         </Item>
</Add>
<Add>
         <CmdID>14</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/Challenge</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>ChallengeInsertedHere</Data>
         </Item>
</Add>
<Add>
         <CmdID>15</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/CAThumbprint</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
            </Meta>
            <Data>CAThumbprintInsertedHere</Data>
         </Item>
</Add>
<Exec>
         <CmdID>16</CmdID>
         <Item>
            <Target><LocURI>./Vendor/MSFT/CertificateStore/My/SCEP/CertSCEP

1/Install/Enroll</LocURI></Target>
         </Item>
```

```
</Exec>
</Atomic>
```

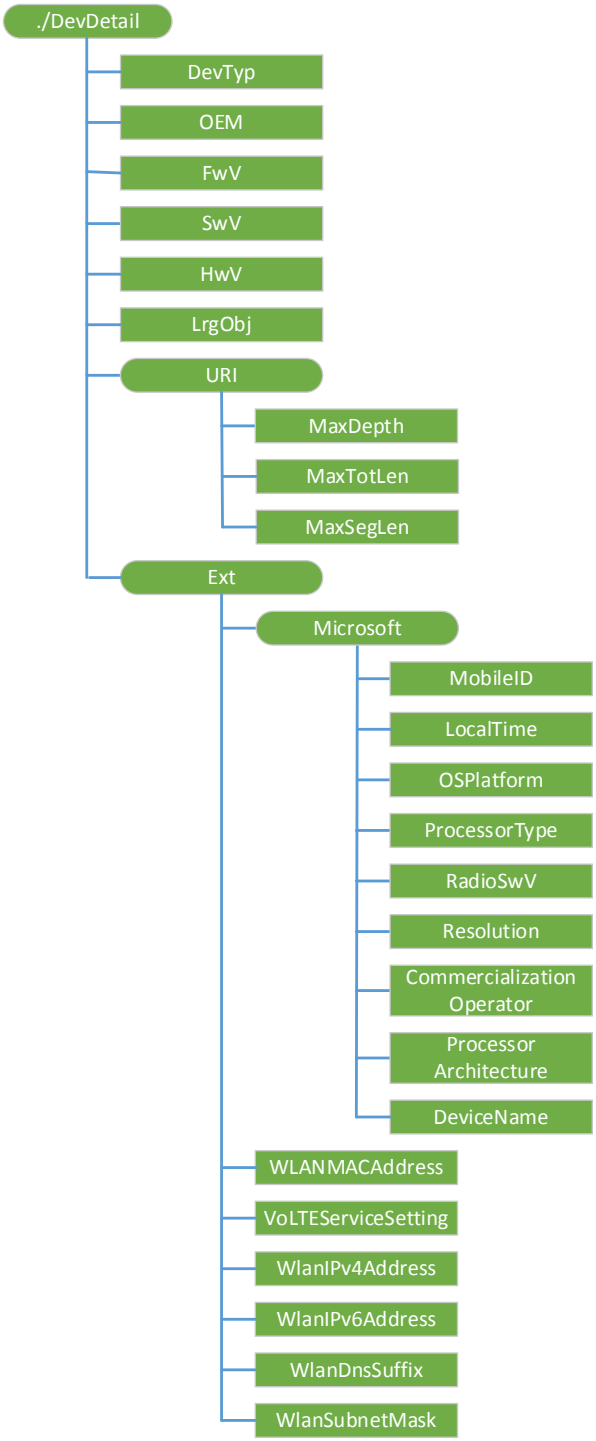
Configuring the device to automatically renew MDM client certificate with specified renew period and retry interval

```
<Atomic>
   <CmdID>1</CmdID>
     <Replace>
         <CmdID>2</CmdID>
         <Item>

<Target><LocURI>./Vendor/MSFT/CertificateStore/My/WSTEP/Renew/ROBOSupport</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">bool</Format>
            </Meta>
            <Data>true</Data>
         </Item>
      </Replace>
      <Replace>
         <CmdID>3</CmdID>
         <Item>

<Target><LocURI>./Vendor/MSFT/CertificateStore/My/WSTEP/Renew/RenewPeriod</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>60</Data>
         </Item>
      </Replace>
      <Replace>
         <CmdID>4</CmdID>
         <Item>

<Target><LocURI>./Vendor/MSFT/CertificateStore/My/WSTEP/Renew/RetryInterval</LocURI></Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>4</Data>
         </Item>
      </Replace>
</Atomic>
```

DevDetail configuration service provider

This CSP is based on the OMA DM standard management object DevDetail; we extend it to provide more useful phone information for the management server.

The following diagram shows the DevDetail configuration service provider management object in tree format. All nodes in this CSP support only the Get command.


./DevDetail
DevTyp
OEM
FwV
SwV
HwV
LrgObj
URI
MaxDepth
MaxTotLen
MaxSegLen
Ext
Microsoft
MobileID
LocalTime
OSPlatform
ProcessorType
RadioSwV
Resolution
Commercialization Operator
Processor Architecture
DeviceName
WLANMACAddress
VoLTEServiceSetting
WlanIPv4Address
WlanIPv6Address
WlanDnsSuffix
WlanSubnetMask

DevTyp
Required. Returns the phone model name as a string.

OEM
Required. Returns the name of the Original Equipment Manufacturer (OEM) as a string, as defined in the specification SyncML Device Information, version 1.1.2.

FwV
Required. Returns the firmware version.

SwV
Required. Returns the Windows Phone OS software version.

HwV
Required. Returns the hardware version.

LrgObj
Required. Returns whether the phone uses OMA DM Large Object Handling, as defined in the specification SyncML Device Information, version 1.1.2. This value is always false.

MaxDepth
Required. Returns the maximum depth of the management tree that the phone supports. The default is "0".

This is the maximum number of URI segments that the phone supports. The default value zero (0) indicates that the phone supports a URI of unlimited depth.

MaxTotLen
Required. Returns the maximum total length of any URI used to address a node or node property. The default is "0".

This is the largest number of characters in the URI that the phone supports. The default value zero (0) indicates that the phone supports a URI of unlimited length.

MaxSegLen
Required. Returns the total length of any URI segment in a URI that addresses a node or node property. The default is "0".

This is the largest number of characters that the phone can support in a single URI segment. The default value zero (0) indicates that the phone supports URI segment of unlimited length.

MobileID
Required. Returns the mobile phone ID associated with the cellular network.

The IMSI value is returned for GSM and UMTS networks. CDMA and worldwide phones will return a 404 Not Found status code error if queried for this element.

LocalTime
Returns the client local time in ISO 8601 format.

OSPlatform
Returns the OS platform of the phone.

ProcessorType
Returns the processor type of the phone.

RadioSwV
Returns the radio stack software version number.

Resolution
Returns the UI screen resolution of the phone (example: "480x800").

CommercializationOperator
Returns the name of the mobile operator.

ProcessorArchitecture
Returns the processor architecture of the phone as "arm" or "x86".

DeviceName
Returns the user-specified phone name.

WLANMACAddress
The MAC address of the active WLAN connection, as a 12-digit hexadecimal number.

VoLTEServiceSetting
This is only exposed to Mobile Operator-based OMA-DM servers. Supported operation is Get.

WlanIPv4Address
The IPv4 address of the active Wi-Fi connection. This is only exposed to Mobile Operator-based OMA-DM servers. Supported operation is Get.

WlanIPv6Address
The IPv6 address of the active Wi-Fi connection. This is only exposed to Mobile Operator-based OMA-DM servers. Supported operation is Get.

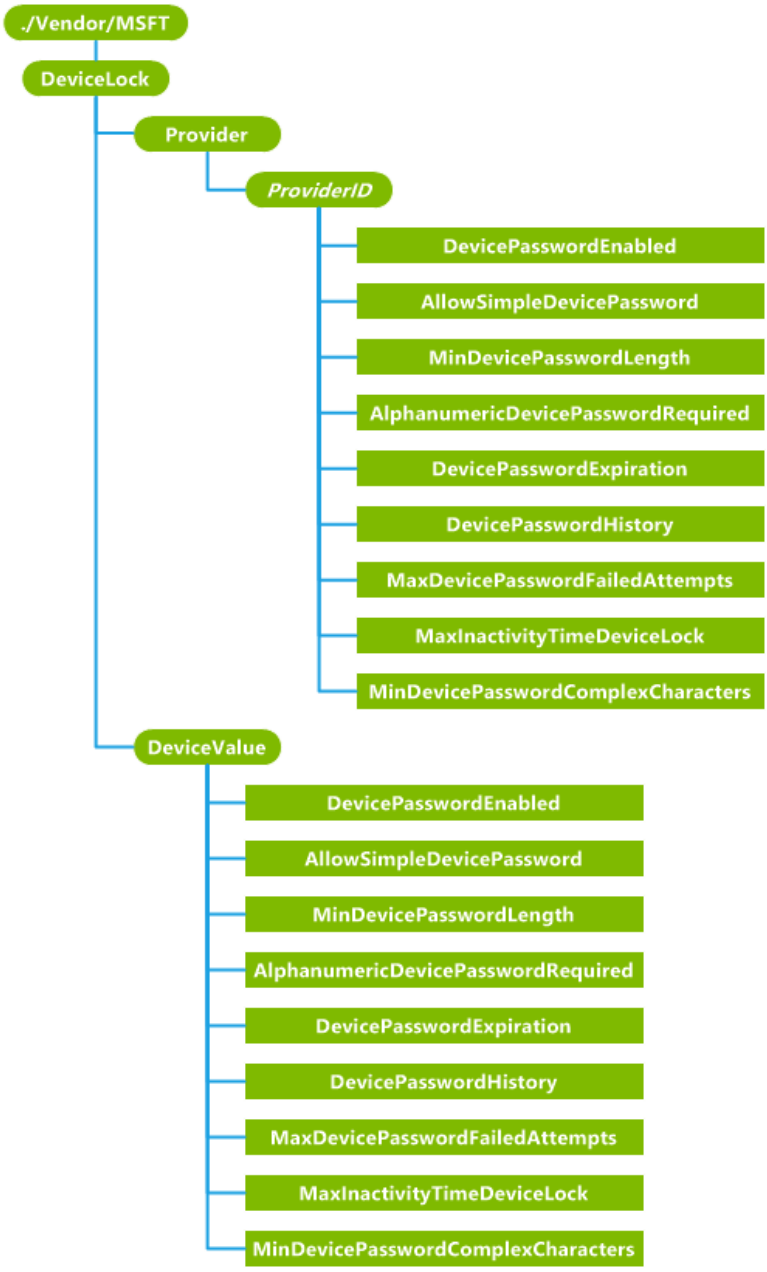
WlanDnsSuffix
The DNS suffix of the active Wi-Fi connection. This is only exposed to Mobile Operator-based OMA-DM servers. Supported operation is Get.

WlanSubnetMask
The subnet mask for the active Wi-Fi connection. This is only exposed to Mobile Operator-based OMA-DM servers. Supported operation is Get.

DeviceLock configuration service provider

This CSP will be deprecated post Windows Phone 8.1. It is recommended to use PolicyManger CSP to configure device lock related policies starting from Windows Phone 8.1.

The following image shows the DeviceLock configuration service provider in tree format.



Provider

Required. An interior node to group all policy providers. Scope is permanent. Supported operation is Get.

Provider/<ProviderID>

Optional. The node that contains the configured management server's ProviderID. In Windows Phone 8, only one enterprise management server is supported. That is, there should be only one ProviderID node. Exchange ActiveSync policies set by Exchange are saved by the Sync client separately. Scope is dynamic. The following operations are supported:

Add

Add the management account to the configuration service provider tree.

Delete

Delete all policies set by this account. This command could be used in enterprise unenrollment for removing policy values set by the enterprise management server.

Get

Return all policies set by the management server.

Note: The value cannot be changed after it is added. The Replace command isn't supported.

/<ProviderID>/DevicePasswordEnabled

Optional. An integer value that specifies whether device lock is enabled. Values are: 1 (device lock not enabled) and 0 (device lock is enabled). An invalid value is treated as configuration failure. The default value is 1. The scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/AllowSimpleDevicePassword

Optional. An integer value that specifies whether simple passwords, such as "1111" or "1234", are allowed. Possible values for this node are: 0 (not allowed), 1 (allowed). Invalid values are treated as a configuration failure. The default value is 1. Scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/MinDevicePasswordLength

Optional. An integer value that specifies the minimum number of characters required in the PIN. Valid values are 4 to 18 inclusive. The default value is 4. Invalid values are treated as a configuration failure. The scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/AlphanumericDevicePasswordRequired

Optional. An integer value that specifies the complexity of the password or PIN allowed. Valid values are 0 (alphanumeric password required), 1 (numeric password required), and 2 (users can choose a numeric password or alphanumeric password). Invalid values are treated as a configuration failure. The scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/DevicePasswordExpiration

Optional. An integer value that specifies the number of days before password expiration. Valid values are 1 to 730. The default value is 0, which indicates that the password does not expire. Invalid values are treated as a configuration failure. The scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/DevicePasswordHistory

Optional. An integer value that specifies the number of passwords that can be stored in the history (can't be reused). Valid values are 0 to 50. The default value is 0. Invalid values are treated as a configuration failure. Scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/MaxDevicePasswordFailedAttempts

Optional. An integer value that specifies the number of authentication failures allowed before the phone will be wiped. Valid values are 0 to 999. The default value is 0, which indicates the phone will

not be wiped regardless of the number of authentication failures. Invalid values are treated as a configuration failure. The scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/MaxInactivityTimeDeviceLock
Optional. An integer value that specifies the amount of time (in minutes) that the phone can remain idle before it is password locked. Valid values are 0 to 999. A value of 0 indicates no time-out is specified. In this case, the maximum screen time-out allowed by the UI applies. Invalid values are treated as a configuration failure. The scope is dynamic. Supported operations are Get, Add, and Replace.

/<ProviderID>/MinDevicePasswordComplexCharacters
Optional. An integer value that specifies the number of complex element types (uppercase and lowercase letters, numbers, and punctuation) required for a strong password. Valid values are 1 to 4. The default value is 1. Invalid values are treated as a configuration failure. Scope is dynamic. Supported operations are Get, Add, and Replace.

DeviceValue
Required. A permanent node that groups the policy values applied to the phone. The server can query this node to discover what policy values are actually applied to the phone. Scope is permanent. Supported operation is Get.

DeviceValue/DevicePasswordEnable, ..., MinDevicePasswordComplexCharacters
Required. This node has the same set of policy nodes as the ProviderID node. All nodes under DeviceValue are read-only permanent nodes. Each node represents the current device lock policy. For detailed descriptions of each policy, see the ProviderID subnode descriptions.

How to implement complex password requirement

When you set **AllowSimpleDevicePassword** to 0 (not allowed) and **AlphanumericDevicePasswordRequired** to 0, the user can still use simple passwords. The work around is to also set **MinDevicePasswordComplexCharacters** to a value greater than 1. The combination the these three settings can prevent the user from using simple passwords, such as 5555.

Example

The following sample shows how to set some device lock policies.

```
<Atomic>
   <CmdID>13</CmdID>
   <Add>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/MaxDevicePasswordFailedAttempts
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>4</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>3</CmdID>
      <Item>
         <Target>
```

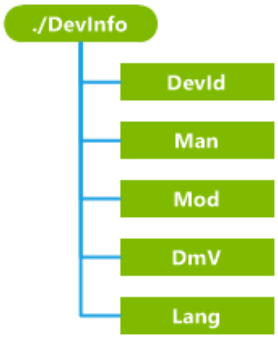
```
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/DevicePasswordEnabled</LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>0</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>4</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/AllowSimpleDevicePassword
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>1</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>5</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/MinDevicePasswordLength
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>5</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>6</CmdID>
      <Item>
         <Target>
            <LocURI>
            ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/AlphanumericDevicePasswordRequired
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>1</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>7</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/DevicePasswordExpiration
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
```

```
        <Data>2</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>8</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/DevicePasswordHistory
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>50</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>9</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/MaxInactivityTimeDeviceLock
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>2</Data>
      </Item>
   </Add>
   <Add>
      <CmdID>10</CmdID>
      <Item>
         <Target>
            <LocURI>
            ./Vendor/MSFT/DeviceLock/Provider/TestMDMServer/MinDevicePasswordComplexCharacters
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>2</Data>
      </Item>
   </Add>
</Atomic>
```

DevInfo configuration service provider

This CSP is based on the OMA DM standard management object DevInfo. It provides some basic phone information to the OMA DM server.

The following diagram shows the DevInfo configuration service provider management object in tree format. All nodes in this CSP support only the Get command.



DevId
Required. Returns an application-specific global unique phone identifier.

Man
Required. Returns the name of the OEM.

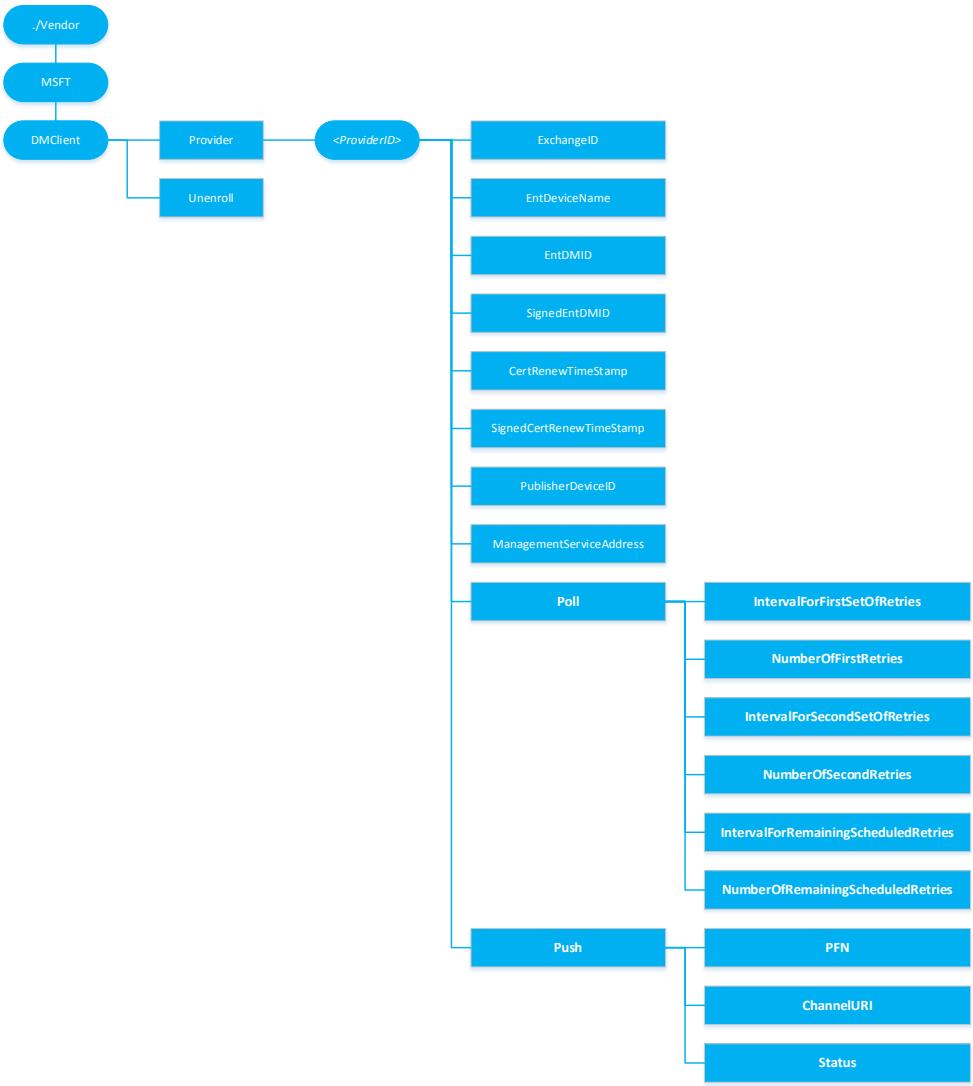
Mod
Required. Returns the name of the hardware phone model as specified by the mobile operator.

DmV
Required. Returns the current management client revision of the phone.

Lang
Required. Returns the current user interface (UI) language setting of the phone as defined by RFC1766.

DMClient configuration service provider (Updated in Windows Phone 8.1)

The following diagram shows the DMClient configuration service provider in tree format.



Unenroll

Required. The node accepts unenrollment requests by way of the OMA DM Exec command and calls the enrollment client to unenroll the phone from the management server whose provider ID is specified in the <Data> tag under the <Item> element. Scope is permanent. Supported operations are Get and Exec.

The following sample SyncML shows how to remotely unenroll the phone. Note that this command should be inserted in general DM packages sent from the server to the phone.

```
<Exec>
   <CmdID>2</CmdID>
   <Item>
      <Target>
```

```
         <LocURI>./Vendor/MSFT/DMClient/Unenroll</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>TestMDMServer</Data>
      <!-- Replace TestMDMServer with the real MDM provider ID value. The value must be
           the same as the one specified in the DMClient/Provider/<ProviderID> node -->
   </Item>
</Exec>
```

Provider

Required. The root node for all settings that belong to a single management server. Scope is permanent. Supported operation is Get.

Provider/<ProviderID>

Required. This node contains the URI-encoded value of the bootstrapped device management account's Provider ID. Scope is dynamic. Supported operations are Get and Add. As a best practice, use text that doesn't require XML/URI escaping.

/<ProviderID>/EntDeviceName

Optional. Character string that contains the user-friendly device name used by the IT admin console. The value is set during the enrollment process by way of the DMClient configuration service provider. You can retrieve it later during an OMA DM session. Supported operations are Get and Add.

/<ProviderID>/EntDMID

Optional. Character string that contains the unique enterprise device ID. The value is set by the management server during the enrollment process by way of the DMClient configuration service provider. You can retrieve it later during an OMA DM session. Supported operations are Get and Add.

Note 1: Although hardware device IDs are guaranteed to be unique, there is a concern that this is not ultimately enforceable during a DM session. The device ID could be changed through the w7 APPLICATION configuration service provider's USEHWDEVID parm by another management server. So during enterprise bootstrap and enrollment, a new device ID is specified by the enterprise server.

Note 2: This node is required and must be set by the server before the client certificate renewal is triggered.

/<ProviderID>/ExchangeID

Optional. Character string that contains the unique Exchange device ID used by the Outlook account. This is useful for the enterprise management server to correlate and merge records for a phone that is managed by exchange and natively managed by a dedicated management server. Supported operation is Get. The following is a Get command sample. *TestMDMServer* should be replaced with actual configured Provider ID.

```
<Get>
   <CmdID>12</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/DMClient/TestMDMServer/ExchangeID</LocURI>
      </Target>
   </Item>
</Get>
```

/<ProviderID>/PublisherDeviceID

Optional. The PublisherDeviceID is a device-unique ID created based on the enterprise Publisher ID. Publisher ID is created based on the enterprise application token and enterprise ID via

./Vendor/MSFT/EnterpriseAppManagement/<enterprise id>/EnrollmentToken. It is to ensure that for one enterprise, each phone has a unique ID associated with it. For the same phone, if it has multiple enterprises' applications, each enterprise is identified differently. Supported operation is Get.

/<ProviderID>/SignedEntDMID

Optional. Character string that contains the device ID. This node and the nodes CertRenewTimeStamp and SignedCertRenewTimeStamp can be used by the mobile device management server to verify client identity in order to update the registration record after the phone certificate is renewed. The phone signs the EntDMID with the old client certificate during the certificate renewal process and saves the signature locally. Supported operation is Get and Replace.

/<ProviderID>/CertRenewTimeStamp

Optional. The time in OMA DM standard time format. This node and the SignedCertRenewTimeStamp node are designed to reduce the risk of the certificate being used by another phone. The phone records the time that the new certificate was created. Supported operation is Get.

/<ProviderID>/SignedCertRenewTimeStamp

Optional. The character string that contains the certificate creation time stamp. The phone signs the certificate creation time stamp with the old certificate immediately after the certificate is renewed. The signature can be retrieved by the server through this configuration service provider. This helps to prevent a man-in-the-middle attack. The signature is valid for approximately 30 minutes. Supported operation is Get.

/<ProviderID>/ManagementServiceAddress

Required. The character string that contains the device management server address. It can be updated during an OMA DM session by the management server to allow the server to load balance to another server in situations where too many devices are connected to the server.

The DMClient configuration service provider will save the address to the same location as the w7 and DMS configuration service providers to ensure the management client has a single place to retrieve the current server address. The initial value for this node is the same server address value as bootstrapped via the w7 APPLICATION configuration service provider.

Supported operations are Get and Replace.

/<ProviderID>/Poll

Optional. Polling schedules in Windows Phone 8.1 must now utilizes the DMClient CSP. The Registry paths previously associated with polling using the Registry CSP are now deprecated. Supported operations are Get, Add.

There are three schedules managed under the Poll node which enable a rich polling schedule experience to provide greater flexibility in managing the way in which devices poll the management server. There are a variety of ways polling schedules may be set. If an invalid polling configuration is set, the device will correct or remove the schedules in order to restore the polling schedules back to a valid configuration.

Valid poll schedule: signoid initial polling schedule with infinite schedule [RECOMMENDED]

Schedule Name	Schedule set by Server	Actual value queried on Device
IntervalForFirstSetOfRetries	15	15
NumberOfFirstRetries	5	5
IntervalForSecondSetOfRetries	60	60
NumberOfSecondRetries	10	10
IntervalForRemainingScheduledRetries	1440	1440

NumberOfRemainingScheduledRetries	0	0

Valid poll schedule: initial enrollment only [no infinite schedule]

Schedule Name	Schedule set by Server	Actual schedule set on Device
IntervalForFirstSetOfRetries	15	15
NumberOfFirstRetries	5	5
IntervalForSecondSetOfRetries	60	60
NumberOfSecondRetries	10	10
IntervalForRemainingScheduledRetries	0	0
NumberOfRemainingScheduledRetries	0	0

Invalid poll schedule: disable all poll schedules

NOTE: Disabling poll schedules results in UNDEFINED behavior and enrollment may fail if poll schedules are all set to zero.

Schedule Name	Schedule set by Server	Actual value queried on Device
IntervalForFirstSetOfRetries	0	0
NumberOfFirstRetries	0	0
IntervalForSecondSetOfRetries	0	0
NumberOfSecondRetries	0	0
IntervalForRemainingScheduledRetries	0	0
NumberOfRemainingScheduledRetries	0	0

Invalid poll schedule: two infinite schedules

Schedule Name	Schedule set by Server	Actual schedule set on Device	Actual Experience
IntervalForFirstSetOfRetries	15	15	Device polls
NumberOfFirstRetries	5	5	
IntervalForSecondSetOfRetries	1440	1440	Device polls server once 24 hrs
NumberOfSecondRetries	0	0	
IntervalForRemainingScheduledRetries	1440	0	Third schedule is disabled
NumberOfRemainingScheduledRetries	0	0	

NOTE 1: If the device was previously MDM enrolled with polling schedule configured via registry key values directly (for example MDM enrolled Windows Phone 8 device upgrade to Windows Phone 8.1), MDM server that supports using DMClient CSP to update polling schedule MUST first send Add command to add ./Vendor/MSFT/DMClient/Provider/<ProviderID>/Poll node before it sends Get/Replace command to query or update polling parameters via DMClient CSP

NOTE 2: when use DMClient CSP to configure polling schedule parameters, the server must not set all 6 polling parameters to 0 or set all 3 number of retries nodes to 0, doing so will cause configuration failure.

/<ProviderID>/Poll/IntervalForFirstSetOfRetries

Optional. The waiting time (in minutes) for the initial set of retries as specified by the number of retries in /<ProviderID>/Poll/NumberOfFirstRetries. If IntervalForFirstSetOfRetries is not set, then the default value is used. The default value is 15. If the value is set to 0, this schedule is disabled.Supported operations are Get, Replace.

Replaces the deprecated HKLM\Software\Microsoft\Enrollment\OmaDmRetry\AuxRetryInterval path that previously utilized the Registry CSP.

/<ProviderID>/Poll/NumberOfFirstRetries

Optional. The number of times the DM client should retry connecting to the server when the client is initially configured/enrolled to communicate with the server. If the value is set to 0 and IntervalForFirstSetOfRetries value isn't 0, then schedule will be set to repeat an infinite number of times and second set and this set of schedule will not set in this case. The default value is 10. Supported operations are Get, Replace.

Replaces the deprecated HKLM\Software\Microsoft\Enrollment\OmaDmRetry\AuxNumRetries path that previously utilized the Registry CSP.

Note that first set of retries is intended to give management server some buffered time to be ready to send policies and settings configuration to the device. The total time for first set of retries shouldn't be more than a few hours. The server shouldn't set NumberOfFirstRetries to be 0. RemainingScheduledRetries is used for long run device polling schedule.

/<ProviderID>/Poll/IntervalForSecondSetOfRetries

Optional. The waiting time (in minutes) for the second set of retries as specified by the number of retries in /<ProviderID>/Poll/NumberOfSecondRetries. Default value is 0. If this value is set to zero, then this schedule is disabled. Supported operations are Get, Replace.

Replaces the deprecated HKLM\Software\Microsoft\Enrollment\OmaDmRetry\RetryInterval path that previously utilized the Registry CSP.

/<ProviderID>/Poll/NumberOfSecondRetries

Optional. The number of times the DM client should retry second round connecting to the server when the client is initially configured/enrolled to communicate with the server. Default value is 0. If the value is set to 0 and IntervalForSecondSetOfRetries isn't set to 0 AND first set of retries isn't set as infinite retries, then schedule will be set to repeat an infinite number of times. However, if first set of retries is set at infinite, then this schedule will be disabled. Supported operations are Get, Replace.

Replaces the deprecated HKLM\Software\Microsoft\Enrollment\OmaDmRetry\NumRetries path that previously utilized the Registry CSP.

Note that second set of retries is also optional and temporarily retries that the total duration should be last for more than a day. And the IntervalForSecondSetOfRetries of should be longer than IntervalForFirstSetOfRetries. RemainingScheduledRetries is used for long run device polling schedule.

/<ProviderID>/Poll/IntervalForRemainingScheduledRetries

Optional. The waiting time (in minutes) for the initial set of retries as specified by the number of retries in /<ProviderID>/Poll/NumberOfRemainingScheduledRetries. Default value is 0. If IntervalForRemainingScheduledRetries is set to 0, then this schedule is disabled. Supported operations are Get, Replace.

Replaces the deprecated HKLM\Software\Microsoft\Enrollment\OmaDmRetry\Aux2RetryInterval path that previously utilized the Registry CSP.

/<ProviderID>/Poll/NumberOfRemainingScheduledRetries

Optional. The number of times the DM client should retry connecting to the server when the client is initially configured/enrolled to communicate with the server. Default value is 0. If the value is set to 0 and IntervalForRemainingScheduledRetries AND first and second set of retries aren't set as infinite retries, then schedule will be set to repeat an infinite number of times. However, if either or both of first and second set of retries are set as infinite, then this schedule will be disabled. Supported operations are Get, Replace.

Replaces the deprecated HKLM\Software\Microsoft\Enrollment\OmaDmRetry\Aux2NumRetries path that previously utilized the Registry CSP.

Note that RemainingScheduledRetries is used for long run device polling schedule. IntervalForRemainingScheduledRetries shouldn't be set small than 1440 minutes (24 hours) in Windows Phone 8.1 device. Windows Phone 8.1 support MDM server push.

/<ProviderID>/Push

Optional. Not configurable during wapprovisioining XML. If removed, DM sessions triggered by Push will no longer be supported. Supported operations are Add, Delete.

/<ProviderID>/Push/PFN

Required. A string provided by the Windows and Windows Phone ecosystem for a Mobile Device Management solution. Used to register a device for Push Notifications. The server must use the same PFN as the devices it is managing. Supported operations are Add, Get, Replace.

/<ProviderID>/Push/ChannelURI

Required. A string that contains the channel that the WNS client has negotiated for the OMA-DM client on the device based on the PFN that was provided. If no valid PFN is currently set, ChannelURI will return null. Supported operation is Get.

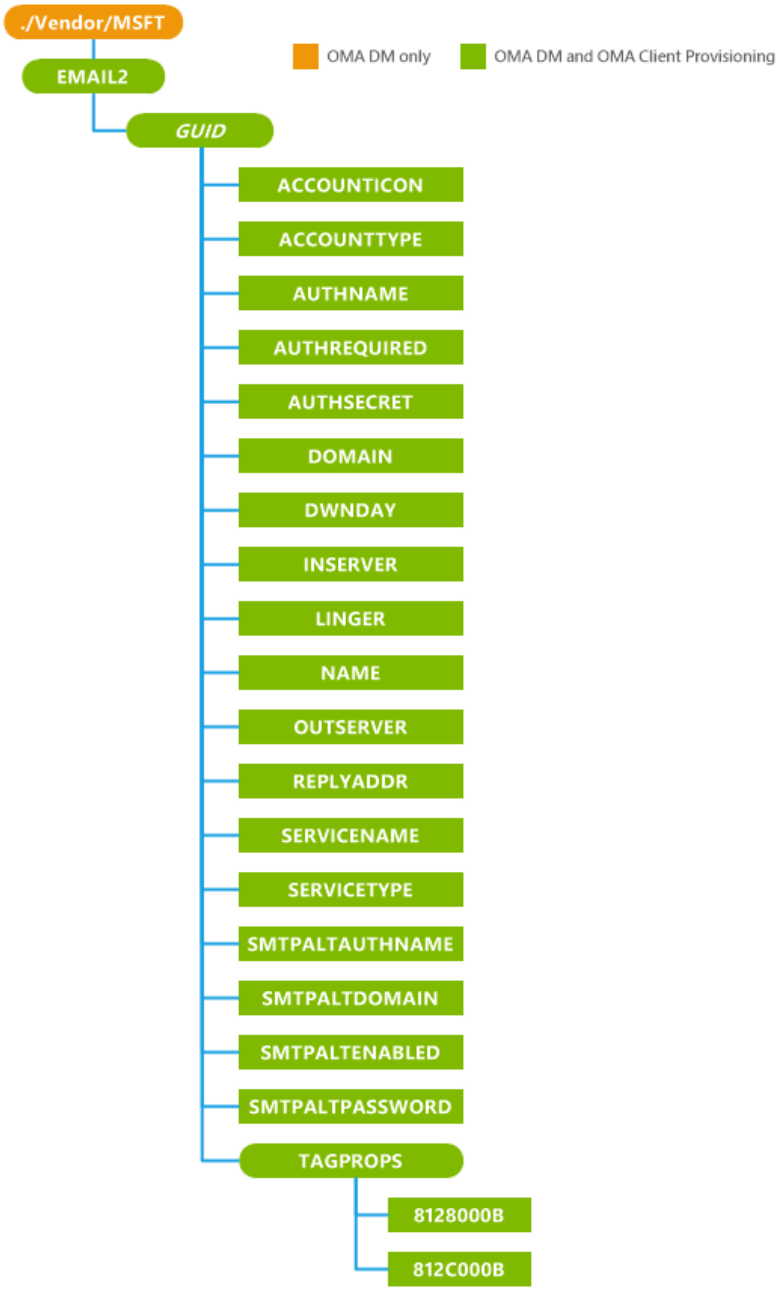
/<ProviderID>/Push/Status

Required. An integer that maps to a known error state or condition on the system. Status error mapping is listed below. Supported operation is Get.

Status	Description
0	Success!
1	Failure: invalid PFN
2	Failure: invalid or expired device authentication with MSA
3	Failure: WNS client registration failed due to an invalid or revoked PFN
4	Failure: no Channel URI assigned
5	Failure: Channel URI has expired
6	Failure: Channel URI failed to be revoked
7	Failure: push notification received, but unable to establish an OMA-DM session due to power or connectivity limitations.
8	Unknown error

EMAIL2 configuration service provider

The following diagram shows the EMAIL2 configuration service provider in tree format.



EMAIL2

The configuration service provider root node. Supported operation is Get.

{GUID}

Defines a specific email account. A globally unique identifier (GUID) must be generated for each email account on the phone - Provisioning with an account that has the same GUID as an existing one does not create the new account and Add command will fail in this case. Supported operations are Get, Add, and Delete.

The braces { } around the GUID are required in the EMAIL2 configuration service provider.

For OMA DM Sync XML, the braces must be sent by using ASCII values of %7B and %7D respectively. For example, <Target><LocURI>./Vendor/MSFT/EMAIL2/**%7B**C556E16F-56C4-4edb-9C64-D9469EE1FBE0**%7D**</LocURI></Target>

ACCOUNTICON

Optional. Returns the location of the icon associated with the account. Supported operations are Get, Add, Replace and Delete.

The account icon can be used as a tile in the Start list or an icon in the applications list under Settings, email & accounts. Some icons are already provided on the phone. The suggested icon for POP/IMAP or generic ActiveSync accounts is at res://AccountSettingsSharedRes{ScreenResolution}!%s.genericmail.png. Custom icons can be added if desired.

For information about adding icons, see Additional accounts on the phone.

ACCOUNTTYPE

Required. Specifies the type of account. Supported operations are: Get and Add. Valid values are:

- Email: normal email
- VVM: visual voice mail

AUTHNAME

Required. Character string specifies the name used to authorize the user to a specific email account (also known as the user's logon name). Supported operations are Get, Add and Replace.

AUTHREQUIRED

Optional. Character string specifies whether the outgoing server requires authentication. Supported operations are Get, Add, Replace and Delete.

A value of "0" specifies that server authentication is not required. A value of "1" specifies that server authentication is required.

AUTHSECRET

Optional. Character string specifies the user's password. The same password is used for SMTP authentication. Supported operations are Get, Add, Replace and Delete.

DOMAIN

Optional. Character string specifies the user's domain name. Supported operations are Get, Add, Replace and Delete.

DWNDAY

Optional. Character string specifies how many days' worth of email should be downloaded from the server. Supported operations are Get, Add, Replace and Delete.

The allowed values are:

- -1: specifies that all email currently on the server should be downloaded.
- 7: specifies that seven days' worth of email should be downloaded
- 14: specifies fourteen days' worth of email should be downloaded.

- 30: specifies thirty days' worth of email should be downloaded.

INSERVER
Required. Character string specifies the name of the messaging service's incoming email server. Supported operations are Get, Add and Replace.

LINGER
Optional. Character string specifies the length of time between email send/receive updates in minutes. The default is 15. Supported operations are Get, Add, Replace and Delete. Allowed values are:

- 0: email updates must be performed manually.
- 15: wait for fifteen minutes
- 30: wait for thirty minutes
- 60: wait for sixty minutes
- 120: wait for one hundred and twenty minutes

NAME
Optional. Character string specifies the name of the sender displayed on a sent email. It should be set to the user's name. Supported operations are Get, Add, Replace and Delete.

OUTSERVER
Required. Character string specifies the name of the messaging service's outgoing email server. Supported operations are Get, Add and Replace.

REPLYADDR
Optional. Character string specifies the user's reply email address (usually the same as the user's email. Send email will fail without it. Supported operations are Get, Add, Delete and Replace.

SERVICENAME
Required. Character string specifies name of the email service to create or edit (32 characters maximum).

The EMAIL2 configuration service provider does not support the OMA DM Replace command on the parameters SERVICENAME and SERVICETYPE. To replace either the email account name or the account service type, the existing email account must be deleted and then a new one must be created. Supported operations are Get, Add, Replace and Delete.

SERVICETYPE
Required. Character string specifies the type of email service to create or edit (for example, "IMAP4" or "POP3").

The EMAIL2 configuration service provider does not support the OMA DM Replace command on the parameters SERVICENAME and SERVICETYPE. To replace either the email account name or the account service type, the existing email account must be deleted and then a new one must be created. Supported operations are Get and Add.

SMTPALTAUTHNAME
Optional. Character string specifies the display name associated with the user's alternative SMTP email account. Supported operations are Get, Add, Replace and Delete.

SMTPALTDOMAIN
Optional. Character string specifies the domain name for the user's alternative SMTP account. Supported operations are Get, Add, Replace and Delete.

SMTPALTENABLED
Optional. Character string specifies if the user's alternate SMPT account is enabled. Supported operations are Get, Add, Replace and Delete.

A value of "FALSE" means SMTP uses the same user name password for authentication. A value of "TRUE" means SMTP uses its own user name password (SMTPALTAUTHNAME and SMTPALTPASSWORD).

SMTPALTPASSWORD
Optional. Character string specifies the password for the user's alternate SMPT account. Supported operations are Get, Add, Replace and Delete.

TAGPROPS
Optional. Defines a group of properties with non-standard element names. Supported operation is Get.

8128000B
Optional. Character string specifies if the incoming email server uses SSL. Supported operations are Get and Replace.

A value of "0" specifies that SSL is not enabled. A value of "1" specifies that SSL is enabled.

812C000B
Optional. Character string specifies if the outgoing email server uses SSL. Supported operations are Get and Replace.

A value of "0" specifies that SSL is not enabled. A value of "1" specifies that SSL is enabled.

Remarks

When an application removal or configuration roll-back is provisioned, the EMAIL2 configuration service provider passes the request to Configuration Manager, which handles the transaction externally. When a MAPI application is removed, the accounts that were created with it are deleted, and all messages and other properties that the transport (for example, Short Message Service [SMS], Post Office Protocol [POP], or Simple Mail Transfer Protocol [SMTP]) might have stored, are lost. If an attempt to create a new email account is unsuccessful, the new account is automatically deleted. If an attempt to edit an existing account is unsuccessful, the original configuration is automatically rolled back (restored).

For OMA DM, the EMAIL2 configuration service provider handles the Replace command differently from most other configuration service providers: Configuration Manager implicitly adds the missing part of the node to be replaced or any segment in the path of the node if it is left out in the <LocURI></LocURI> block. There are separate parameters defined for the outgoing server logon credentials. The following are the usage rules for these credentials:

- The incoming server logon credentials are used (AUTHNAME, AUTHSECRET, and DOMAIN) unless the outgoing server credentials are set.
- If some but not all of the outgoing server credentials parameters are present, the EMAIL2 configuration service provider will be considered in error.

The phone supports Transport Layer Security (TLS), but this cannot be explicitly enabled through this configuration service provider, and the user cannot enable TLS through the UI. If the connection to the mail server is initiated with deferred SSL, the mail server can send STARTTLS as a server capability and TLS will be enabled. The following steps show how TLS can be enabled.

1. The phone attempts to connect to the mail server using SSL.
2. If the SSL connection fails, the phone attempts to connect using deferred SSL.
3. If the connection fails over both SSL and deferred SSL, and the user selected Server requires encrypted (SSL) connection, the phone does not attempt another connection.
4. If the user did not select Server requires encrypted (SSL) connection, the phone attempts to establish a non-SSL connection.

5. If the connection succeeds using any of the encryption protocols, the phone requests the server capabilities.
6. If one of the capabilities sent by the mail server is STARTTLS and the connection is deferred SSL, the phone enables TLS. TLS is not enabled on connections using SSL or non-SSL.

When managing over OMA DM, make sure to always use a unique GUID. Provisioning with an account that has the same GUID as an existing one deletes the existing account and does not create the new account.

Examples

IMAP account configuration

The following sample shows how to use SyncML commands to configure an IMAP email account. It must be wrapped in a SyncML package sent from the server. The GUID must be replaced with the appropriate unique GUID.

```
<Atomic>
   <CmdID>1</CmdID>
   <Add>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">node</Format>
         </Meta>
      </Item>
   </Add>
   <Replace>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/SERVICENAME
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>ExampleIMAP</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>3</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/SERVICETYPE
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>IMAP4</Data>
      </Item>
   </Replace>
   <Replace>
```

```
      <CmdID>4</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/REPLYADDR
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>user@contoso.com</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>5</CmdID>
      <Item>
         <Target>
            <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/NAME
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>Contoso IMAP</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>6</CmdID>
      <Item>
         <Target>
            <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/INSERVER
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>imap.contoso.com</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>7</CmdID>
      <Item>
         <Target>
            <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/DOMAIN
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data></Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>8</CmdID>
         <Item>
         <Target>
            <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/AUTHNAME
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
```

```
     <Data>user</Data>
  </Item>
</Replace>
<Replace>
   <CmdID>9</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/AUTHSECRET
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>ThisIsAPassword</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>10</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/OUTSERVER
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>smtp.mail.contoso.com</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>11</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/AUTHREQUIRED
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>1</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>12</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/SMTPALTENABLED
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
     <Data>0</Data>
  </Item>
</Replace>
<Replace>
   <CmdID>13</CmdID>
   <Item>
      <Target>
         <LocURI>
```

```
            ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/SMTPALTDOMAIN
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data></Data>
   </Item>
</Replace>
<Replace>
   <CmdID>14</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/SMTPALTAUTHNAME
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data></Data>
   </Item>
</Replace>
<Replace>
   <CmdID>15</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/SMTPALTPASSWORD
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data></Data>
   </Item>
</Replace>
<Replace>
   <CmdID>16</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/LINGER
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>120</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>17</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/DWNDAY
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>7</Data>
   </Item>
```

```
</Replace>
<Replace>
   <CmdID>18</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/KEEPMAX
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>0</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>19</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/RETRIEVE
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>20480</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>20</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/ACCOUNTTYPE
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>Email</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>23</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/TAGPROPS
         </LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
      </Meta>
   </Item>
</Replace>
<Replace>
   <CmdID>24</CmdID>
   <Item>
      <Target>
         <LocURI>
          ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/TAGPROPS/8128000B
         </LocURI>
      </Target>
      <Data>1</Data>
```

```
      </Item>
   </Replace>
   <Replace>
      <CmdID>25</CmdID>
      <Item>
         <Target>
            <LocURI>
             ./Vendor/MSFT/EMAIL2/%7B1BC45B68-A51F-4AF1-B6C1-BC22746DAE82%7D/TAGPROPS/812C000B
            </LocURI>
         </Target>
         <Data>1</Data>
      </Item>
   </Replace>
</Atomic>
```

POP3 account configuration

The following sample shows how to use SyncML commands to configure a POP3 email account. It must be wrapped in a SyncML package sent from the server. The GUID must be replaced with an appropriate unique GUID.

```
<Atomic>
   <CmdID>3</CmdID>
   <Add>
      <CmdID>2</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">node</Format>
            <Type xmlns="syncml:metinf">text/plain</Type>
         </Meta>
      </Item>
   </Add>
   <Replace>
      <CmdID>5</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/SERVICENAME
            </LocURI>
         </Target>
         <Data>Service1</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>6</CmdID>
      <Item>
         <Target>
            <LocURI>
               ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/SERVICETYPE
            </LocURI>
         </Target>
         <Data>POP3</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>7</CmdID>
      <Item>
```

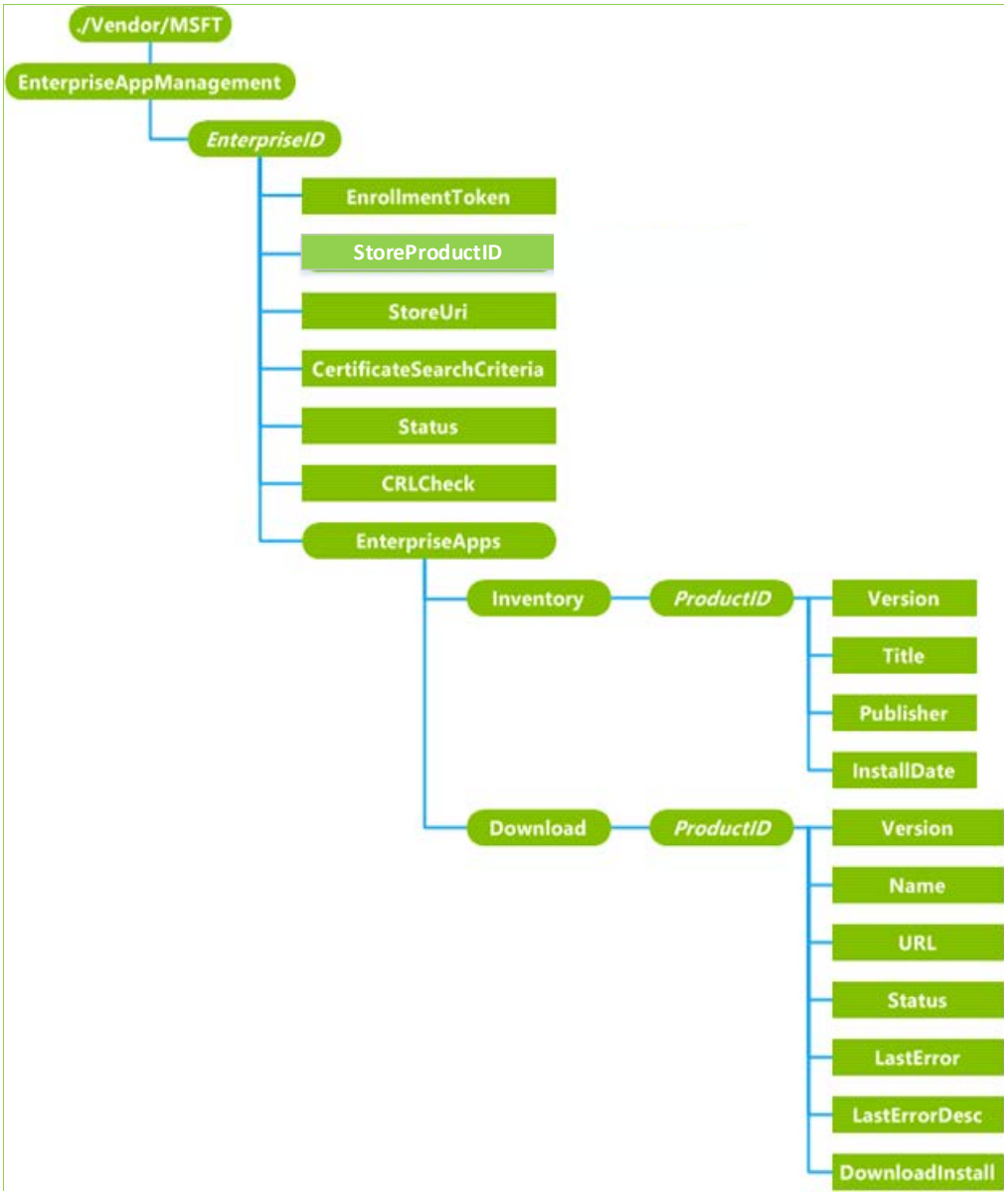
```
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/REPLYADDR
         </LocURI>
      </Target>
      <Data>user@contoso.com</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>8</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/NAME
         </LocURI>
      </Target>
      <Data>test1</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>9</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/INSERVER
         </LocURI>
      </Target>
      <Data>pop.contoso.com</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>11</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/AUTHNAME
         </LocURI>
      </Target>
      <Data>user@contoso.com</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>13</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e048b25e87216%7D/OUTSERVER
         </LocURI>
      </Target>
      <Data>smtp.contoso.com</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>14</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/AUTHREQUIRED
         </LocURI>
      </Target>
      <Data>0</Data>
   </Item>
```

```
</Replace>
<Replace>
   <CmdID>15</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/SMTPALTENABLED
         </LocURI>
      </Target>
      <Data>0</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>21</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/RETRIEVE
         </LocURI>
      </Target>
      <Data>2048</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>22</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/KEEPMAX
         </LocURI>
      </Target>
      <Data>0</Data>
   </Item>
</Replace>
<Add>
   <CmdID>24</CmdID>
   <Item>
      <Target>
         <LocURI>
            ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/TAGPROPS
         </LocURI>
      </Target>
   </Item>
</Add>
<Replace>
   <CmdID>25</CmdID>
   <Item>
      <Target>
         <LocURI>
          ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/TAGPROPS/8128000B
         </LocURI>
      </Target>
      <Data>1</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>26</CmdID>
   <Item>
      <Target>
         <LocURI>
          ./Vendor/MSFT/EMAIL2/%7B4ebb5cb3-e382-4104-a04e-048b25e87216%7D/TAGPROPS/812C000B
         </LocURI>
```

```
        </Target>
        <Data>0</Data>
      </Item>
   </Replace>
</Atomic>
```

EnterpriseAppManagement configuration service provider (added functionality for Windows Phone 8.1)

The following diagram shows the EnterpriseAppManagement configuration service provider in tree format.



{EnterpriseID}

Optional. A dynamic node that represents the EnterpriseID as a GUID. It is used to enroll or unenroll enterprise applications. Supported operations are Add, Delete, and Get.

{EnterpriseID}/EnrollmentToken

Required. Used to install or update the binary representation of the application enrollment token (AET) and initiate "phone home" token validation. Scope is dynamic. Supported operations are Get, Add, and Replace.

{EnterpriseID}/StoreProductID

Required. The string that contains the ID of the first enterprise application (usually a Company Hub app), which is automatically installed on the phone. Scope is dynamic. Supported operations are Get and Add.

{EnterpriseID}/StoreUri

Optional. The character string that contains the URI of the first enterprise application to be installed on the phone. The enrollment client downloads and installs the application from this URI. Scope is dynamic. Supported operations are Get and Add.

{EnterpriseID}/CertificateSearchCriteria

Optional. The character string that contains the search criteria to search for the DM-enrolled client certificate. The certificate is used for client authentication during enterprise application download. The company's application content server should use the enterprise-enrolled client certificate to authenticate the phone. The value must be a URL encoded representation of the X.500 distinguished name of the client certificates Subject property. The X.500 name must conform to the format required by CertStrToName (refer to http://msdn.microsoft.com/en-us/library/windows/desktop/aa377160(v=vs.85).aspx). This search parameter is case sensitive. Scope is dynamic. Supported operations are Get and Add.

NOTE: Do NOT use Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00. The server must replace this value in the supplied client certificate. If your server returns a client certificate containing the same Subject value, this can cause unexpected behavior. The server should always override the subject value and not use the default device-provided Device ID Subject= Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00

{EnterpriseID}/Status

Required. The integer value that indicates the current status of the application enrollment. Valid values are 0 (ENABLED), 1 (INSTALL_DISABLED), 2 (REVOKED), and 3 (INVALID). Scope is dynamic. Supported operation is Get.

{EnterpriseID}/CRLCheck

Optional. Character value that specifies whether the phone should do a CRL check when using a certificate to authenticate the server. Valid values are "1" (CRL check required), "0" (CRL check not required). Scope is dynamic. Supported operations are Get, Add, and Replace.

{EnterpriseID}/EnterpriseApps

Required. The root node to for individual enterprise application related settings. Scope is dynamic (this node is automatically created when EnterpriseID is added to the configuration service provider). Supported operation is Get.

/EnterpriseApps/Inventory

Required. The root node for individual enterprise application inventory settings. Scope is dynamic (this node is automatically created when EnterpriseID is added to the configuration service provider). Supported operation is Get.

/Inventory/ProductID

Optional. A node that contains s single enterprise application product ID in GUID format. Scope is dynamic. Supported operation is Get.

/Inventory/ProductID/Version

Required. The character string that contains the current version of the installed enterprise application. Scope is dynamic. Supported operation is Get.

/Inventory/ProductID/Title
Required. The character string that contains the name of the installed enterprise application. Scope is dynamic. Supported operation is Get.

/Inventory/ProductID/Publisher
Required. The character string that contains the name of the publisher of the installed enterprise application. Scope is dynamic. Supported operation is Get.

/Inventory/ProductID/InstallDate
Required. The time (in the character format YYYY-MM-DD-HH:MM:SS) that the application was installed or updated. Scope is dynamic. Supported operation is Get.

/EnterpriseApps/Download
Required. This node groups application download-related parameters. Note that for Windows Phone 8, the enterprise server can only automatically update currently installed enterprise applications. The end user controls which enterprise applications to download and install. Scope is dynamic. Supported operation is Get.

/Download/ProductID
Optional. This node contains the GUID for the installed enterprise application. Each installed application has a unique ID. Scope is dynamic. Supported operations are Get, Add, and Replace.

/Download/ProductID/Version
Optional. The character string that contains version information (set by the caller) for the application currently being downloaded. Scope is dynamic. Supported operations are Get, Add, and Replace.

/Download/ProductID/Name
Required. The character string that contains the name of the installed application. Scope is dynamic. Supported operation is Get.

/Download/ProductID/URL
Optional. The character string that contains the URL for the updated version of the installed application. The phone will download application updates from this link. Scope is dynamic. Supported operations are Get, Add, and Replace.

/Download/ProductID/Status
Required. The integer value that indicates the status of the current download process. The following table shows the possible values.

0: CONFIRM	Waiting for confirmation from user
1: QUEUED	Waiting for download to start
2: DOWNLOADING	In the process of downloading
3: DOWNLOADED	Waiting for installation to start
4: INSTALLING	Handed off for installation
5: INSTALLED	Successfully installed
6: FAILED	Application was rejected (not signed properly, bad XAP format, not enrolled properly, etc.)
7:DOWNLOAD_FAILED	unable to connect to server, file doesn't exist, etc.

Scope is dynamic. Supported operation is Get.

/Download/ProductID/LastError

Required. The integer value that indicates the HRESULT of the last error code. If there are no errors, the value is 0 (S_OK). Scope is dynamic. Supported operation is Get.

/Download/ProductID/LastErrorDesc

Required. The character string that contains the human readable description of the last error code.

/Download/ProductID/DownloadInstall

Required. The character string that contains the command to the phone to trigger the download and installation. The server must query the phone later to determine the status. For each product ID, the status field is retained for up to one week. Scope is dynamic. Supported operation is Exec.

Remarks

Install and Update Line of Business (LOB) applications

A workplace can automatically install and update Line of Business applications during a management session. Line of Business applications support a variety of file types including XAP (8.0 and 8.1), AppX, and AppXBundles. A workplace can also update applications from XAP file formats to Appx and AppxBundle formats through the same channel. See the Examples section below for more detailed information.

Uninstall Line of Business (LOB) applications

A workplace can also remotely uninstall Line of Business applications on the device. See the Examples section below for more detailed information. It is not possible to use this mechanism to uninstall Store applications on the device or Line of Business applications that are not installed by the enrolled workplace (for side-loaded application scenarios).

Query installed Store application

It is possible to query if a Store application is installed on a system. First, the Store application GUID should be known. This can be discovered by going to windowsphone.com and finding the application GUID in the URL for the Store application.

For example, the Microsoft Store application has a WindowsPhone.com URL of:

http://www.windowsphone.com/en-us/store/app/microsoft-store/d5dc1ebb-a7f1-df11-9264-00237de2db9e

The Microsoft Store application has a GUID of **d5dc1ebb-a7f1-df11-9264-00237de2db9e**

Use the following syncml format to query to see if the application is installed on a managed device:

```
    <Get>
      <CmdID>1</CmdID>
      <Item>
        <Target>

<LocURI>./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7B
D5DC1EBB-A7F1-DF11-9264-00237DE2DB9E%7D</LocURI>
        </Target>
      </Item>
    </Get>
```

Response from the phone (it contains list of subnodes if this app is installed in the device)

```
<Results>
   <CmdID>3</CmdID>
   <MsgRef>1</MsgRef>
```

```
   <CmdRef>2</CmdRef>
   <Item>
      <Source>
          <LocURI>
             ./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7B
D5DC1EBB-A7F1-DF11-9264-00237DE2DB9E%7D</LocURI>
      </Source>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
         <Type xmlns="syncml:metinf"></Type>
      </Meta>
       <Data>Version/Title/Publisher/InstallDate</Data>
   </Item>
</Results>
```

Node Values

All node values under the ProviderID interior node represent the policy values that the management server wants to set:

- An Add or Replace command on those nodes returns success in both of the following cases:
  - The value is actually applied to the phone.
  - The value isn't applied to the phone because the phone has a more secure value set already.

  From a security perspective, the phone complies with the policy request that is at least as secure as the one requested.
- A Get command on those nodes returns the value the server pushes down to the phone.
- If a Replace command fails, the node value is set to be the previous value before Replace command was applied.
- If an Add command fails, the node is not created.

The value actually applied to the phone can be queried via the nodes under the DeviceValue interior node.

Examples

The following samples show how the EnterpriseAppManagment CSP is used for various scenarios.

Enroll an Enterprise Enrollment Token

Enroll enterprise ID "4000000001" for the first time in SyncML format.

```
<Add>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnrollmentToken</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>InsertTokenHere</Data>
   </Item>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EnterpriseAppManagement/4000000001/CertificateSearchCriteria
         </LocURI>
      </Target>
      <Meta>
```

```
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>SearchCriteriaInsertedHere</Data>
   </Item>
</Add>
```

Update enrollment token

Update the enrollment token (for example, to update an expired application enrollment token)

```
<Replace>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnrollmentToken</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">chr</Format>
      </Meta>
      <Data>InsertUpdaedTokenHere</Data>
   </Item>
</Replace>
```

Query installed applications

Query all installed applications that belong to enterprise id "4000000001"

```
<Get>
   <CmdID>2</CmdID>
   <Item>
      <Target>
          <LocURI>
    ./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory?list=StructData
          </LocURI>
      </Target>
   </Item>
</Get>
```

Response from the phone (it contains two installed applications)

```
<Results>
   <CmdID>3</CmdID>
   <MsgRef>1</MsgRef>
   <CmdRef>2</CmdRef>
   <Item>
      <Source>
          <LocURI>
             ./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory
          </LocURI>
      </Source>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
         <Type xmlns="syncml:metinf"></Type>
      </Meta>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB316008A-141D-
4A79-810F-8B764C4CFDFB%7D
         </LocURI>
      </Source>
      <Meta>
```

```
         <Format xmlns="syncml:metinf">node</Format>
         <Type xmlns="syncml:metinf"></Type>
      </Meta>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D
         </LocURI>
      </Source>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
         <Type xmlns="syncml:metinf"></Type>
      </Meta>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/Version
         </LocURI>
      </Source>
      <Data>1.0.0.0</Data>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/Title
         </LocURI>
      </Source>
      <Data>Sample1</Data>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/Publisher
         </LocURI>
      </Source>
      <Data>ExamplePublisher</Data>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/InstallDate
         </LocURI>
      </Source>
      <Data>2012-10-30T21:09:52Z</Data>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/Version
         </LocURI>
      </Source>
      <Data>1.0.0.0</Data>
   </Item>
   <Item>
```

```
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/Title
         </LocURI>
      </Source>
      <Data>Sample2</Data>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/Publisher
         </LocURI>
      </Source>
      <Data>Contoso</Data>
   </Item>
   <Item>
      <Source>
         <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB0322158-C3C2-
44EB-8A31-D14A9FEC450E%7D/InstallDate
         </LocURI>
      </Source>
      <Data>2012-10-31T21:23:31Z</Data>
   </Item>
</Results>
```

Install and Update an enterprise application

Install or Update the installed app with product ID "{B316008A-141D-4A79-810F-8B764C4CFDFB}"

To perform an XAP update, create the Name, URL, Version, and DownloadInstall nodes first, then perform an "execute" on the "DownloadInstall" node (all within an "Atomic" operation). If the application does not exist, the application will be silently installed without any user interaction. If the application cannot be installed, the user will be notified with an Alert dialog.

NOTE1: that if a previous app-update node existed for this product ID (the node can persist for up to 1 week or 7 days after an installation has completed), then a 418 (already exist) error would be returned on the "Add". To get around the 418 error, the server should issue a Replace command for the Name, URL, and Version nodes, and then execute on the "DownloadInstall" (within an "Atomic" operation).

NOTE2: the application product ID curly braces need to be escaped where { is %7B and } is %7D

```
<Atomic>
   <CmdID>2</CmdID>
   <!-- The Add command can be used if the download node does not have a matching product ID
        node in it or application was installer 7 or more days old.Otherwise, use the Replace
command. -->
   <Add>
      <CmdID>3</CmdID>
      <Item>
         <Target>
            <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Download/%7BB316008A-141D-
4A79-810F-8B764C4CFDFB%7D/Name
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
```

```
         </Meta>
         <Data>ContosoApp1</Data>
      </Item>
      <Item>
         <Target>
            <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Download/%7BB316008A-141D-
4A79-810F-8B764C4CFDFB%7D/URL
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>http://contoso.com/enterpriseapps/ContosoApp1.xap</Data>
      </Item>
      <Item>
         <Target>
            <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Download/%7BB316008A-141D-
4A79-810F-8B764C4CFDFB%7D/Version</LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <Data>2.0.0.0</Data>
      </Item>
      <Item>
         <Target>
            <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Download%7BB316008A-141D-4A79-
810F-8B764C4CFDFB%7D/DownloadInstall
            </LocURI>
         </Target>
         <Data>1</Data>
      </Item>
   </Add>
   <Exec>
      <CmdID>4</CmdID>
      <Item>
         <Target>
            <LocURI>
./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Download/%7BB316008A-141D-
4A79-810F-8B764C4CFDFB%7D/DownloadInstall
            </LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <Data>0</Data>
      </Item>
   </Exec>
</Atomic>
```

Uninstall enterprise application

Uninstall an installed enterprise application with product ID "{7BB316008A-141D-4A79-810F-8B764C4CFDFB }"

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Delete>
      <CmdID>2</CmdID>
      <Item>
```

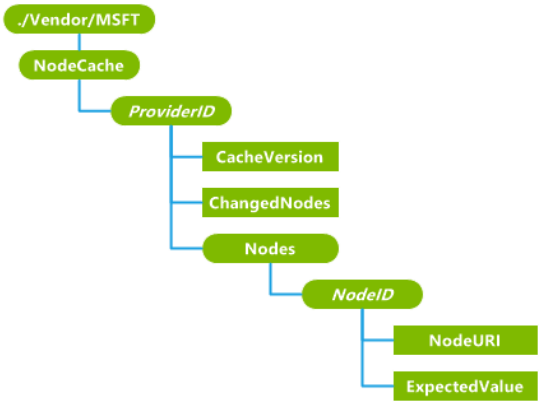
```
        <Target>

<LocURI>./Vendor/MSFT/EnterpriseAppManagement/4000000001/EnterpriseApps/Inventory/%7BB316008A-
141D-4A79-810F-8B764C4CFDFB%7D</LocURI>
        </Target>
      </Item>
    </Delete>
    <Final/>
  </SyncBody>
</SyncML>
```

NodeCache configuration service provider

The following diagram shows the NodeCache configuration service provider in tree format. The management could leverage this CSP to ask the device to track CSP node value change by setting NodeURI to be monitored. By checking ChangedNodes node, the server will know which node's value is changed.

Note 1: Using NodeCache CSP to track WiFi profile update isn't supported.

Note 2: Node cache information is deleted when the device is upgrade from one version to another version. The MDM server should re-establish nodes to be tracked including adding all nodes under NodeCache CSP root node including ProviderID node.



NodeCache

Required. The root node for the NodeCache object. In Windows Phone 8.1, this configuration service provider is used for enterprise device management only. Supported operation is Get. This is a predefined MIME type to identify this managed object in OMA DM syntax. The value in Windows Phone 8.1 is: com.microsoft/1.0/WindowsPhone/NodeCache

<ProviderID>

Optional. Group settings per DM server. Each group of settings is distinguished by the server's Provider ID. It should be the same DM server ProviderID value that was supplied through the w7 APPLICATION configuration service provider XML during the enrollment process. In Windows Phone 8, only one enterprise management server is supported. That is, there should be only one ProviderID node under NodeCache. Scope is dynamic. Supported operations are Get, Add, and Delete.

<ProviderID>/CacheVersion

Optional. Character string value is set by the server when the set of nodes or their expected values changes. Scope is dynamic. Supported operations are Get, Add, and Replace.

<ProviderID>/ChangedNodes
Optional. List of nodes whose values do not match their expected values as specified in /<NodeID>/ExpectedValue. Scope is dynamic. Supported operation is Get.

<ProviderID>/Nodes
Required. Root node for cached nodes. Scope is dynamic. Supported operation is Get.

/Nodes/<NodeID>
Optional. Information about each cached node is stored under <NodeID> as specified by the server. This value must not contain a comma. Scope is dynamic. Supported operations are Get, Add, and Delete.

/<NodeID>/NodeURI
Required. This node's value is a complete OMA DM node URI. It can specify either an interior or leaf node in the device management tree. Scope is dynamic. Supported operations are Get, Add, and Delete.

/<NodeID>/ExpectedValue
Required. This is the value that the server expects to be on the phone. When the configuration service provider initiates a session, it checks the expected value against the node's actual value. Scope is dynamic. Supported operations are Get, Add, and Delete.

Remarks

Typical OMA DM Session with the NodeCache CSP

1. Phone connects to a DM server.
2. Server queries the NodeCache version by issuing a GET operation for ./Vendor/MSFT/NodesCache/<*ProviderID*>/CacheVersion LocURI
3. If the phone's CacheVersion and the server-side cache differ (due to reasons such as a phone crash or server crash), the server can clear the server-side cache and go to step 5.
4. Server updates the server-side cache:
    a. Sends GET operation for ./Vendor/MSFT/NodeCache/<*ProviderID*>/ChangedNodes LocURI
    b. Response is a list of changed node IDs. Each ID in the list corresponds to a node under ./Vendor/MSFT/NodeCache/<*ProviderID*>/Nodes root
    c. For each node in the invalid nodes list, server sends a GET command to retrieve the actual value of the node:
    GET <NodeURI>
    (where NodeURI is a full device LocURI that correspond to the invalid cache node.)
    d. Nodes in the server-side cache are updated with the actual values received from the phone.
    e. For each updated node, a REPLACE command is sent to the phone to update the phone-side cache:
    REPLACE ./Vendor/MSFT/NodesCache/<*ProviderID*>/Nodes/<*NodeID*>/ExpectedValue => ActualValue
    f. A new cache version is created and sent to the phone:
    REPLACE ./Vendor/MSFT/NodesCache/<*ProviderID*>/CacheVersion => new_version
    The new_version value is stored by the server
5. Management server retrieves the corresponding value from the server-side cache:

a. If a value already exists in the server-side cache, retrieve the value from server-side cache instead of going to the phone.

b. If a value does not exist in the server-side cache, then:

   i. Create a new entry with a unique <*NodeID*> in the server side cache.

   ii. Query the phone to retrieve the actual value of the URI.

   iii. Create a new node under ./Vendor/MSFT/NodesCache/<*ProviderID*>/Nodes with <*NodeID*> value.

   iv. Set up NodeURI and ExpectedValue for the ./Vendor/MSFT/NodesCache/<*ProviderID*>/Nodes/<*NodeID*> node.

   v. Update the CachedNodes version.

Examples

Creating settings for node caching

```
<Add>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
      </Meta>
   </Item>
</Add>
<Add>
   <CmdID>4</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0001</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
      </Meta>
   </Item>
</Add>
<Add>
   <CmdID>5</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0001/NodeURI</LocURI>
      </Target>
      <Data>./Vendor/MSFT/DeviceLock/Provider/MDMSRV1/DevicePasswordEnabled</Data>
   </Item>
</Add>
<Add>
   <CmdID>6</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0001/ExpectedValue</LocURI>
      </Target>
      <Data>0</Data>
   </Item>
</Add>
<Add>
   <CmdID>8</CmdID>
```

```
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0002</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">node</Format>
      </Meta>
   </Item>
</Add>
<Add>
   <CmdID>9</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0002/NodeURI</LocURI>
      </Target>
      <Data>
         ./Vendor/MSFT/DeviceLock/Provider/MDMSRV1/AlphanumericDevicePasswordRequired
      </Data>
   </Item>
</Add>
<Add>
   <CmdID>10</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0002/ExpectedValue</LocURI>
      </Target>
      <Data>0</Data>
   </Item>
</Add>
```

Getting nodes under Provider ID MDMSRV1, cache version, changed nodes, node, expected value

```
<Get>
   <CmdID>18</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1</LocURI>
      </Target>
   </Item>
</Get>
<Get>
   <CmdID>19</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/CacheVersion</LocURI>
      </Target>
   </Item>
</Get>
<Get>
   <CmdID>20</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/SCCM01/ChangedNodes</LocURI>
      </Target>
   </Item>
</Get>
<Get>
   <CmdID>21</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDM1/Nodes/Node_0001</LocURI>
      </Target>
   </Item>
```

```
</Get>
<Get>
   <CmdID>22</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/SCCM01/Nodes/Node_0001/ExpectedValue</LocURI>
      </Target>
   </Item>
</Get>
```

Replacing the cache version, node URI, and expected value

```
<Replace>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/CacheVersion</LocURI>
      </Target>
      <Data>SCCM0001@!Replace</Data>
   </Item>
</Replace>
<Replace>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0001/NodeURI</LocURI>
      </Target>
      <Data>./Vendor/MSFT/DeviceLock/DeviceValue/AllowSimpleDevicePassword</Data>
    </Item>
</Replace>
<Replace>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/NodeCache/MDMSRV1/Nodes/Node_0001/ExpectedValue</LocURI>
      </Target>
      <Data>2</Data>
   </Item>
</Replace>
```

RemoteWipe configuration service provider

The following diagram shows the RemoteWipe configuration service provider management object in tree format as used by DM client.

doWipe

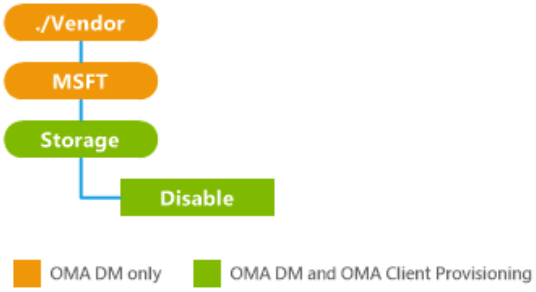
Specifies that a remote wipe of the phone should be performed.

When used with OMA Client Provisioning, a dummy value of "1" should be included for this element.

Supported operation is Exec.

Storage configuration service provider

The following diagram shows the Storage configuration service provider in tree format. This CSP will be deprecated post Windows Phone 8.1. It is recommended to use PolicyManger CSP to configure storage card policy starting from Windows Phone 8.1.



Disable

Required. Specifies whether to enable or disable a storage card. A Boolean value of true disables the storage card. The default value is False. The value is case sensitive.

The supported operations are Get and Replace.

Note that if the phone returns a 404 error code when the server applies the Get command to ./Vendor/MSFT/Storage/Disable, it means that the phone doesn't have an SD card.

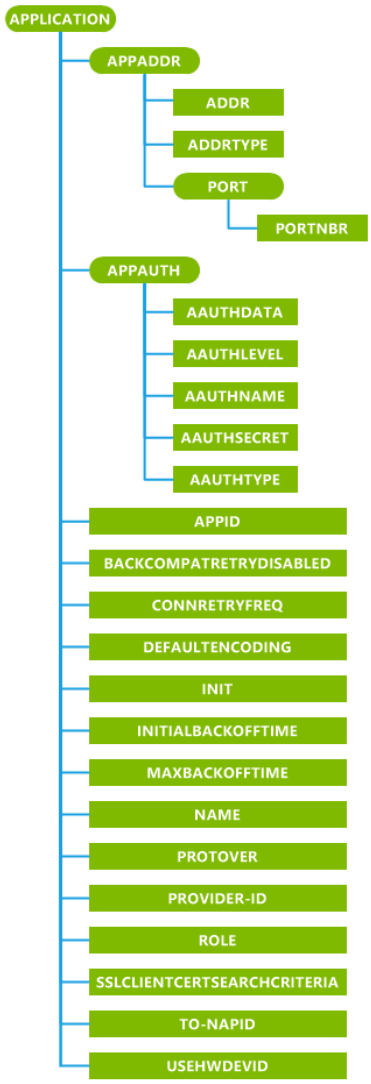
w7 APPLICATION configuration service provider

The APPLICATION configuration service provider that has an APPID of w7 is used for bootstrapping a phone with an OMA DM account. Although this configuration service provider is used to set up an OMA DM account, it is managed over OMA Client Provisioning.

NOTE 1: All parm name and characteristic types are case sensitive and must use all uppercase.

NOTE 2: Both APPSRV and CLIENT credentials must be provided in provisioning XML.

The following image shows the configuration service provider in tree format as used by OMA DM.



APPADDR

This characteristic is used in the w7 APPLICATION characteristic to specify the DM server address.

APPADDR/ADDR

Optional. The ADDR parameter is used in the APPADR characteristic to get or set the address of the OMA DM server. This parameter takes a string value.

APPADDR/ADDRTYPE

Optional. The ADDRTYPE parameter is used in the APPADDR Characteristic to get or set the format of the ADDR parameter. This parameter takes a string value.

In OMA DM XML, if there are multiple instances of this parameter, the first valid parameter value is used.

APPADDR/PORT

This characteristic is used in the APPADDR characteristic to specify port information.

APPADDR/PORT/PORTNBR

Required. The PORTNBR parameter is used in the PORT characteristic to get or set the number of the port to connect to. This parameter takes a numeric value in string format.

APPAUTH

This characteristic is used in the w7 APPLICATION characteristic to specify authentication information.

APPAUTH/AAUTHDATA

Optional. The AAUTHDATA parameter is used in the APPAUTH characteristic to get or set additional data used in authentication. This parameter is used to convey the nonce for MD5 digest authentication type. This parameter takes a string value. The value of this parameter is base64-encoded in the form of a series of bytes. Note that if the AAUTHTYPE is DIGEST, this is used as a nonce value in the MD5 hash calculation, and the octal form of the binary data should be used when calculating the hash at server side and device side.

APPAUTH/AAUTHLEVEL

Required. The AAUTHLEVEL parameter is used in the APPAUTH characteristic to indicate whether credentials are for server authentication or client authentication. This parameter takes a string value. You can set this value.

The valid values are listed below:

- APPSRV specifies that the *client* authenticates itself to the OMA DM Server at the DM protocol level.
- CLIENT specifies that the *server* authenticates itself to the OMA DM Client at the DM protocol level.

APPAUTH/AAUTHNAME

Optional. The AAUTHNAME parameter is used in the APPAUTH characteristic to differentiate OMA DM client names. This parameter takes a string value. You can set this value.

APPAUTH/AAUTHSECRET

Required. The AAUTHSECRET parameter is used in the APPAUTH characteristic to get or set the authentication secret used to authenticate the user. This parameter takes a string value.

APPAUTH/AAUTHTYPE

Optional. The AAUTHTYPE parameter of the APPAUTH characteristic is used to set the method of authentication. This parameter takes a string value.

The valid values are listed below:

- BASIC specifies that the SyncML DM 'syncml:auth-basic' authentication type.
- DIGEST specifies that the SyncML DM 'syncml:auth-md5' authentication type.

- When AAUTHLEVEL is CLIENT, AAUTHTYPE must be DIGEST. When AAUTHLEVEL is APPSRV, AAUTHTYPE can be BASIC or DIGEST.

APPID

Required. The APPID parameter is used in the APPLICATION characteristic to differentiate the types of available application services and protocols. This parameter takes a string value. The only valid value to configure the OMA Client Provisioning bootstrap APPID is w7.

BACKCOMPATRETRYDISABLED

Optional. The BACKCOMPATRETRYDISABLED parameter is used in the APPLICATION characteristic to specify whether to retry resending a package with an older protocol version (for example, 1.1) in the SyncHdr (not including the first time).

NOTE: This parameter doesn't contain a value. The existence of this parameter means backward compatibility retry is disabled. If the parameter is missing, it means backward compatibility retry is enabled.

CONNRETRYFREQ

Optional. The CONNRETRYFREQ parameter is used in the APPLICATION characteristic to specify how many retries the DM client performs when there are Connection Manager–level or WinInet-level errors. This parameter takes a numeric value in string format. The default value is "3". You can set this parameter.

DEFAULTENCODING

Optional. The DEFAULTENCODING parameter is used in the APPLICATION characteristic to specify whether the DM client should use WBXML or XML for the DM package when communicating with the server. You can get or set this parameter. The valid values are:

- application/vnd.syncml.dm+xml (Default)
- application/vnd.syncml.dm+wbxml

INIT

Optional. The INIT parameter is used in the APPLICATION characteristic to indicate that the management server wants the client to initiate a management session immediately after settings approval.

INITIALBACKOFFTIME

Optional. The INITIALBACKOFFTIME parameter is used in the APPLICATION characteristic to specify the initial wait time in milliseconds when the DM client retries for the first time. The wait time grows exponentially. This parameter takes a numeric value in string format. The default value is "16000". You can set this parameter.

MAXBACKOFFTIME

Optional. The MAXBACKOFFTIME parameter is used in the APPLICATION characteristic to specify the maximum number of milliseconds to sleep after package-sending failure. This parameter takes a numeric value in string format. The default value is "86400000". You can set this parameter.

NAME

Optional. The NAME parameter is used in the APPLICATION characteristic to specify a user readable application identity. This parameter is used to define part of the registry path for the APPLICATION parameters. You can set this parameter.

The NAME parameter can be a string or null (no value). If no value is specified, the registry location will default to <unnamed>.

PROTOVER

Optional. The PROTOVER parameter is used in the APPLICATION characteristic to specify the OMA DM Protocol version the server supports. No default value is assumed. The protocol version set by this node will match the protocol version that the DM client reports to the server in SyncHdr in package 1. If this node is not specified when adding a DM server account, the latest DM protocol version that the client supports is used. In Windows Phone this is 1.2. This is a Microsoft custom parameter. You can set this parameter.

PROVIDER-ID

Optional. The PROVIDER-ID parameter is used in the APPLICATION characteristic to differentiate OMA DM servers. It specifies the server identifier for a management server used in the current management session. This parameter takes a string value. You can set this parameter.

ROLE

Optional. The ROLE parameter is used in the APPLICATION characteristic to specify the security application chamber the DM session should run with when communicating with the DM server. For enterprise management, role value 32 (Enterprise) is supported. This is a Microsoft custom parameter. This parameter takes a numeric value in string format.

TO-NAPID

Optional. The TO-NAPID parameter is used in the APPLICATION characteristic to specify the Network Access Point the client will use to connect to the OMA DM server. If multiple TO-NAPID parameters are specified, only the first TO-NAPID value will be stored. This parameter takes a string value. You can set this parameter.

USEHWDEVID

Optional. The USEHWDEVID parameter is used in the APPLICATION characteristic to specify use of phone hardware identification. It does not have a value.

- If the parameter is not present, the default behavior is to use an application-specific GUID used rather than the hardware device ID.
- If the parameter is present, the hardware device ID will be provided at the ./DevInfo/DevID node and in the Source LocURI for the DM package sent to the server. International Mobile Station Equipment Identity (IMEI) is returned for a GSM phone.

SSLCLIENTCERTSEARCHCRITERIA

Optional. The SSLCLIENTCERTSEARCHCRITERIA parameter is used in the APPLICATION characteristic to specify the client certificate search criteria. This parameter supports search by subject attribute and certificate stores. If any other criteria are provided, it is ignored.

The string is a concatenation of name/value pairs, each member of the pair delimited by the "&" character. The name and values are delimited by the "=" character. If there are multiple values, each value is delimited by the Unicode character "U+F000". If the name or value contains characters not in the UNRESERVED set (as specified in RFC2396), then those characters are URI-escaped per the RFC.

The supported names are Subject and Stores; wildcard certificate search isn't supported.

Stores specifies which certificate stores the DM client will search to find the SSL client certificate. The valid store value is My%5CUser. The store name is not case sensitive.

Note: %EF%80%80 is the UTF8-encoded character U+F000.

Subject specifies the certificate to search for. For example, to specify that you want a certificate with a particular Subject attribute ("CN=Tester,O=Microsoft"), use the following:

```
<parm name="SSLCLIENTCERTSEARCHCRITERIA"
```

```
   value="Subject=CN%3DTester,O%3DMicrosoft&amp;amp;Stores=My%5CUser" />
```

NOTE: Do NOT use Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00. The server must replace this value in the supplied client certificate. If your server returns a client certificate containing the same Subject value, this can cause unexpected behavior The server should always override the subject value and not use the default device-provided Device ID Subject= Subject=CN%3DB1C43CD0-1624-5FBB-8E54-34CF17DFD3A1\x00

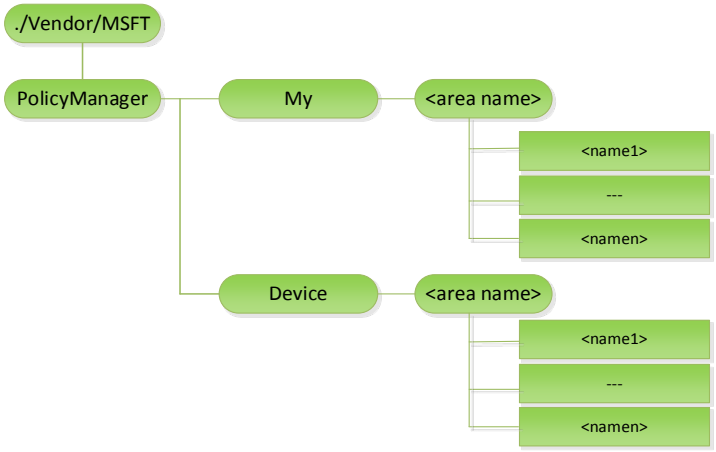
PolicyManager configuration service provider (New in Windows Phone 8.1)

In Windows Phone 8.1, PolicyManager is the centralized component to handle all Windows Phone supported enterprise policies: for both new policies added in Windows Phone 8.1 and previous policies supported by Windows Phone 8 and handled by other CSPs. It is recommended that MDM server should use this centralized CSP to configure any company policies.

The CSP has two major sub categories: PolicyManager/My/<Area>/<policy name> path handles the policy configuration request coming from the server, PolicyManager/Device/<Area>/<policy name> is read only path to reflect the policy values that are enforced at the device.

Please note that configuration of policies for the same <Area> must be wrapped within Atomic command. Refer Examples section for some sample xmls.

The following diagram shows the PolicyManager configuration service provider in tree format.



./Vendor/MSFT/**PolicyManager**

- Description: Policy Manager CSP root node.
- Format: node
- Supported operations: Get
- Occurrence: One
- Type: This is a predefined MIME type to identify this managed object in OMA DM syntax. com.microsoft/1.0/WindowsPhone/PolicyManagerMO

./Vendor/MSFT/PolicyManager/**My**

- Description: An interior node that indicates that policies provisioned by a specific provider are to be retrieved, modified, or deleted.
- Format: node
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/PolicyManager/My/**<area name>**

- Description: An interior node grouping all policies that can be configured by a single technology for a single provider. Once added, the value cannot be changed, e.g. Replace command isn't supported.
- Format: node
- Supported operations: Add, Get, Delete
- Occurrence: OneOrMore

./Vendor/MSFT/PolicyManager/My/<area name>/**<name>**

- Description: The node specifies a name/value pair used in the policy. Note that for multi strings value, it will be separated by a specific Unicode  in the XML file. The multi strings will be terminated with  : One stringtwo stringred stringWindows Phone 8.1 string. Note that a query from different caller could provide a different value as each caller could have different values for named policy.
    - N**OTE**: Any Syncml used to set policy should be wrapped with the Atomic XML tag which treats the policy settings as a single transaction.
- Format: string
- Supported operations: Add, Get, Delete, Replace
- Occurrence: OneOrMore.

./Vendor/MSFT/PolicyManager/**Device**

- Description: An interior node grouping all the evaluated policies that can be configured. This node corresponds to the evaluated policies of all the providers.
- Format: node
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/PolicyManager/Device/**<area name>**

- Description: An interior node grouping all policies that can be configured by a single technology independent of the providers.
- Format: node
- Supported operations: Get
- Occurrence: OneOrMore

./Vendor/MSFT/PolicyManager/Device/<area name>/**<name>**

- Description: The node specifies a name/value pair used in the policy.
- Format: string
- Supported operations: Get
- Occurrence: OneOrMore.

Windows Phone 8.1 supported company policies

The following table shows Windows Phone 8.1 company policies that are configurable by MDM server add/or Exchange servers.

Area/Policy name	Description	Supported value	Value evaluation rule	Supported via MDM or EAS	EAS policy name
DeviceLock /DevicePasswordEnabled	Specifies whether device lock is enabled.	1 (default) - Not required 0 – Required	Min (policy values) is most restricted value	MDM, EAS	DevicePasswordEnabled
DeviceLock /AllowSimpleDevicePassword	Specifies if password like "1111" or "1234" are allowed.	0 - Not allowed 1 (default) – Allowed	Min (policy values) is most restricted value	MDM, EAS	AllowSimpleDevicePassword
DeviceLock /MinDevicePasswordLength	Specifies the minimum number or characters. required in the PIN	An integer X where 4 <= X <= 16 0- not enforced Default: 4	Max (policy values) is most restricted value	MDM, EAS	MinDevicePasswordLength
DeviceLock /AlphanumericDevicePasswordRequired	Determines the type of password required. This policy only applies if DevicedPasswordEnabled policy is set to 0 (required).	0 - Alphanumeric password required 1 - numeric password required 2 (default) - users can choose: Numeric Password, or Alphanumeric Password	Min (policy values) is most restricted value	MDM, EAS	AlphanumericDevicePasswordRequired
DeviceLock /DevicePasswordExpiration	Specifies when the password expires (in days).	An integer X where 0 <= X <= 730 0 (default) - Passwords do not expire	If all policy values = 0 then 0, else Min (policy values) is most secure value	MDM, EAS	DevicePasswordExpiration
DeviceLock /DevicePasswordH	Specifies how many	An integer X where	Max (policy values) is	MDM, EAS	DevicePasswordHistory

istory	passwords can be stored in the history that can't be used.	0 <= X <=50 Default: 0	most restricted value		
DeviceLock /MaxDevicePasswordFailedAttempts	The number of authentication failures before the device will be wiped. A value of 0 disables device wipe functionality.	An integer X where 0 <= X <= 999 Default: 0 (device will not get wiped after enter any times of wrong password)	If all policy values = 0 then 0, else Min (policy values) is most restricted value	MDM, EAS	MaxDevicePasswordFailedAttempts
DeviceLock /MaxInactivityTimeDeviceLock	Specifies the amount of time (in minutes) after the device is idle that will cause the device to become password locked.	An integer X where 0 <= X <= 999 0 (default) - No timeout is defined. The default of "0" is Mango parity and is interpreted by as "No timeout is defined."	Min (policy values) (except '0') is most restricted value	MDM, EAS	MaxInactivityTimeDeviceLock
DeviceLock /MinDevicePasswordComplexCharacters	The number of complex element types (uppercase and lowercase letters, numbers, and punctuation) required for a strong password.	An integer X where 1 <= X <= 4 Default: 1	Max (policy values) is most restricted value	MDM, EAS	MinDevicePasswordComplexCharacters
WiFi /AllowWiFi	Allow or disallow WiFi connection. (Configurable by Exchange as well –	0 – use WiFi connection is disallowed 1 (default) – use Wi-Fi	Most restricted value is 0	MDM, EAS	AllowWiFi

	definition will be consistent with EAS definition.)	connection is allowed.			
WiFi /AllowInternetSharing	Allow or disallow internet sharing (Configurable by Exchange as well – definition will be consistent with EAS definition.)	0 – Do not allow the use of Internet Sharing. 1 (default) – Allow the use of Internet Sharing	Most restricted value is 0	MDM, EAS	AllowInternetSharing
WiFi /AllowAutoConnectToWiFiSenseHotspots	Allow or disallow the device to automatically connect to Wi-Fi hotspots and friend social network.	0 – not allowed 1 (default) – allowed	Most restrict value is 0	MDM	
WiFi /AllowWiFiHotSpotReporting	Allow or disallow WiFi Hotspot information reporting to Microsoft. Once disallowed, the user cannot turn it on.	0 – HotSpot reporting is not allowed 1 (default) – HotSpot reporting is allowed	Most restricted value is 0	MDM	
WiFi /AllowManualWiFiConfiguration	Allow or disallow connecting to Wi-Fi outside of MDM server-installed networks.	0 – no WiFi connection outside of MDM provisioned is allowed 1 (default) – adding new network SSIDs beyond the already	Most restricted value is 0	MDM	

		MDM provisioned ones is allowed			
Connectivity /AllowNFC	Allow or disallow NFC. Only MDM server can set it.	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM	
Connectivity /AllowBluetooth	Set Bluetooth mode. (Could be set by Exchange EAS policy as well, definition be the same as Exchange).	0 – disallow Bluetooth 1 (not supported in Windows Phone 8.1) – Disable Bluetooth, but allow the configuration of hands-free profiles – NOTE: value 1 isn't supported in Windows Phone 8.1 for MDM and EAS. 2 (default) – allow Bluetooth	Secure order (Most restricted value is 0) 0 2	MDM, EAS	AllowBluetooth
Connectivity /AllowBluetoothSharing (new for GDR2)	Set to allow Bluetooth sharing.	0 – do not allow Bluetooth sharing 1 (default) – allow Bluetooth sharing.			
Connectivity /AllowVPNRoamingOverCellular	This policy when enforced, will prevent the device from connecting VPN when the device roams	0 – not allowed 1 (default) allowed	Most restricted value is 0	MDM	

	over cellular networks.				
Connectivity **/AllowVPNOverCellular**	This policy specifies what type of underline connections VPN is allowed to use	0 - VPN is not allowed over cellular 1 (default) – VPN could use any connection including cellular.	Most restricted value is 0	MDM	
Connectivity/AllowManualVPNConfiguration (new for GDR2)	This policy allows the enterprise to enforce a VPN protection by disabling all VPN settings. It prevents the user from manually configuring VPN settings that does not comply with company security policy.	0 – All VPN settings are disabled for end user from device side. 1(Default) – all VPN settings are enabled for user from device side.	Most restricted value is 0	MDM	
Connectivity **/CellularAppDownloadMBLimit (new for GDR2)**	This policy specifies the maximum app file size in MB allowed for downloading through celluar connection	0 - Boolean (value of "0" interpreted as 20MB 1 - interpreted as mobile operator imposed limit. Default value "0".		MDM	
Wifi/WLANScanMode (new for GDR2)	This policy defines the frequency mode for active Wi-Fi scanning trigger when screen is off and on. High setting would	Integer policy; 0 – Default, 100 – normal interval 500 – low interval		MDM	

	result in faster/better WiFi discoverbility	Default is 0, but 0 interpreted as normal interval			
System /AllowStorageCard	Disable/enable SD card.	0 – SD cared use is not allowed 1 (default) – SD cared use is allowed	Most restricted value is 0	MDM, EAS	AllowStorageCard
System /AllowTelemetry	Allow the device to send telemetry information (such as SQM, Watson).	0 – not allowed 1 – allowed, except for Secondary Data Requests 2 (default) – allowed	Most restricted value is 0	MDM	
Experience /AllowCopyPaste	Specify whether copy and paste is allowed.	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM	
Experience/AllowTaskSwitcher (new for GDR2)	This policy allows the company to disable the task switcher competely. It does not effect the back button action, just the visual switcher trigger by the hold back button action.	0 – disable task switcher 1(Default) – enable task switcher	Mos restricted value is 0	MDM	
Accounts /AllowMicrosoftAccountConnection	Specify whether allow using MSA account for non email related connection authentication	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM	

	and services.				
Accounts /AllowAddingNonMicrosoftAccountsManually	Specify whether user is allowed to add non MSA email accounts.	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM	
Security /AllowManualRootCertificateInstallation	Specify whether the user is allowed to manually install root and intermediate CAP certificates.	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM	
Security /RequireDeviceEncryption	Allow enterprise to turn on internal storage encryption. Note that once turned on, it cannot be turned off via policy.	0 (default) – encryption is not required. 1 – encryption is required.	Most restricted value is 1	MDM, EAS	RequireDeviceEncryption
Security/AntiTheftMode (new for GDR2)	Allows enterprise to preventing user from enabling the Anti Theft mode. Note, if user already enabled the Anti Theft mode for the device before the policy applied, they will have to manually disable the Anti Theft mode for this policy to take effect	0 – do not allow Anti Theft mode to be enabled. 1 – (Default) allow anti-theft mode.	Most restricted value is 0	MDM	
ApplicationManag	Specify	0 – not	Most	MDM	

ement/AllowStore	whether app store is allowed at the device.	allowed 1 (default) – allowed	restricted value is 0		
ApplicationManagement/ApplicationRestrictions	A xml blob specify the application restrictions company want to put to the device. It could be app allow list, app disallow list, allowed publisher IDs, etc. Refer Enterprise application restrictions section on how to set application restriction policy in details. NOTE: the application may not be immediately terminated if the application was previously running	Chr	The information for PolicyManager is opaque. PolicyManager doesn't do most restricted value evaluation. Whenever there is a change to the value, the device parses the node value and enforce the restriction policies specified in the policy	MDM	
ApplicationManagement/AllowDeveloperUnlock	Specify whether developer unlock is allowed at the device.	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM	
Browser/AllowBrowser	Specify whether IE is allowed in the device.	0 – not allowed 1 (default) – allowed	Most restricted value is 0	MDM, EAS	AllowBrowser
Experience/AllowScreenCaptu	Specify whether	0 – not allowed	Most restricted	MDM	

re	screen capture is allowed.	1 (default) – allowed	value is 0		
System /AllowLocation	Allow/Disallow location service.	0 – not allowed 1 (default) – allowed 2 –When set, the location service is always turned on. The **Settings > Location** in the user interface is disabled and the location services toggle will be turned on. The following message is displayed to the user: "Enabled by company policy."	Most restricted value is 0	MDM	
Connectivity /AllowUSBConnection	Allow/Disallow desktop to access phone storage via USB. (Both MTP and IPoUSB) are disabled when policy enforced.	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Connectivity /AllowCellularDataRoaming	Allow or disallow cellular data roaming	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Camera /AllowCamera	Disable/Enable camera	0 – Use camera is disallowed	Most restricted value is 0	MDM	

		1 (default) – Use camera is allowed			
Search/AllowSearchToUseLocation	Specify whether search could leverage location information.	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Search/SafeSearchPermissions **(not supported)**	Specify what level of safe search (filtering adult content) is required. **Note:** This is not supported in Windows Phone 8.1	0 – Strict, highest filtering against adult content 1 (default) – Moderate, moderate filtering against adult content (valid search results will not be filtered)	Most restricted value is 0	MDM	
Search/AllowStoringImagesFromVisionSearch	Specify whether allow BingVision to store the contents of the images captured when performing Bing Vision search	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Experience/AllowVoiceRecording	Specify whether voice recording is allowed	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Experience/AllowSaveAsOfOfficeFiles	Specify whether the user is allowed to save file in the device as office file.	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	

	Note that this policy is for the Office Hub only.				
Experience/AllowSharingOfOfficeFiles	Specify whether the user is allowed to share office file. Note that this policy is for the Office Hub only.	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
AboveLock/AllowActionCenterNotifications	Specify whether allow action center notifications above the device lock screen	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
DeviceLock/AllowIdleReturnWithoutPassword	Force user to input password every time the device is returning from idle state	0 - user is not able to set the password grace period timer, and the value is set as “each time” 1 (default) - user is able to set the password grace period timer	Most restricted value is 0	MDM	
Experience/AllowCortana	Specify whether Cortana is allowed at the device	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Experience /AllowSyncMySettings	Allow enterprise to disallow roaming settings among devices (in/from WP	0 – roaming is disallowed 1(default) – enterprise don’t enforce disallow	Most restricted value is 0	MDM	

	device). If not enforced, whether roaming is allowed or not could depend on other factors. depends on other factors.	roaming			
DataProtection/RequireProtectionUnderLockConfig (new for GDR2)	Allows data encryption of email data and associated attachments. Pin lock key is required to unlock and decode the content.	0(default) – data protection under lock is disabled 1 – data protection under lock is enabled	Most restriced value is 1	MDM	
DataProtection/EnterpriseProtectedDomainNames (new for GDR2)	Specifies the enterprise domain names	String – domain name. Multiple domain names may be defined using "|" character as the separator. Example Contoso.com|Fabrikam.com Default value: <empty>			

Company Owned/Provided/Liable Device Policies

WARNING

This feature should only be used on devices that are owned or provided by the enterprise company or organization or on a user owned device where the user allowed the device to be fully managed by the enterprise company.

As a Mobile Device Management Solutions Vendor, you must provide the following disclaimer to the IT administrator prior to the use of the feature.

This feature may cause the device to fail or lose connectivity and require that the device be serviced at a Nokia-authorized repair center to reset to factory settings. Microsoft is not liable for any damage to the device or any loss of productivity that results from use of this feature. Microsoft requires that software vendors provide disclaimers to users when their products expose this feature and capabilities.

Area/Policy name	Description	Supported value	Value evaluation rule	Supported via MDM or EAS	EAS policy name
System/AllowUserToResetPhone	Specify whether allow the user to factory reset the phone from setting control panel and hardware key combination	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	
Experience/AllowManualMDMUnenrollment	Specify whether allow the user to delete the workplace account via workplace control panel. The MDM server always could remotely delete the account.	0 – not allowed 1(default) – allowed	Most restricted value is 0	MDM	

Examples

Disable Internet sharing and manual Wi-Fi configuration

```
<Atomic>
    <CmdID>1</CmdID>
    <Replace>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/PolicyManager/My/WiFi/AllowInternetSharing</LocURI>
        </Target>
```

```
        <Data>0</Data>
      </Item>
     </Replace>
    <Replace>
      <CmdID>3</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/PolicyManager/My/WiFi/AllowManualWiFiConfiguration
</LocURI>
        </Target>
        <Data>0</Data>
      </Item>
     </Replace>
</Atomic>
```

Query to find out what Camera policy value is applied to the device.

This is important in case multiple resource (such as Exchange servers and MDM server) could configure this policy.

```
<Get>
     <CmdID>2</CmdID>
     <Item>
 <Target>
<LocURI>./Vendor/MSFT/PolicyManager/Device/Camera/AllowCamera</LocURI>
       </Target>
      </Item>
</Get>
```

VPN configuration service provider (New in Windows Phone 8.1)

Windows Phone 8.1 supports both IKEv2 VPN and SSL VPN profiles. Refer http://technet.microsoft.com/en-us/library/ff687731%28v=ws.10%29.aspx for server IKEv2/IPSec configuration.

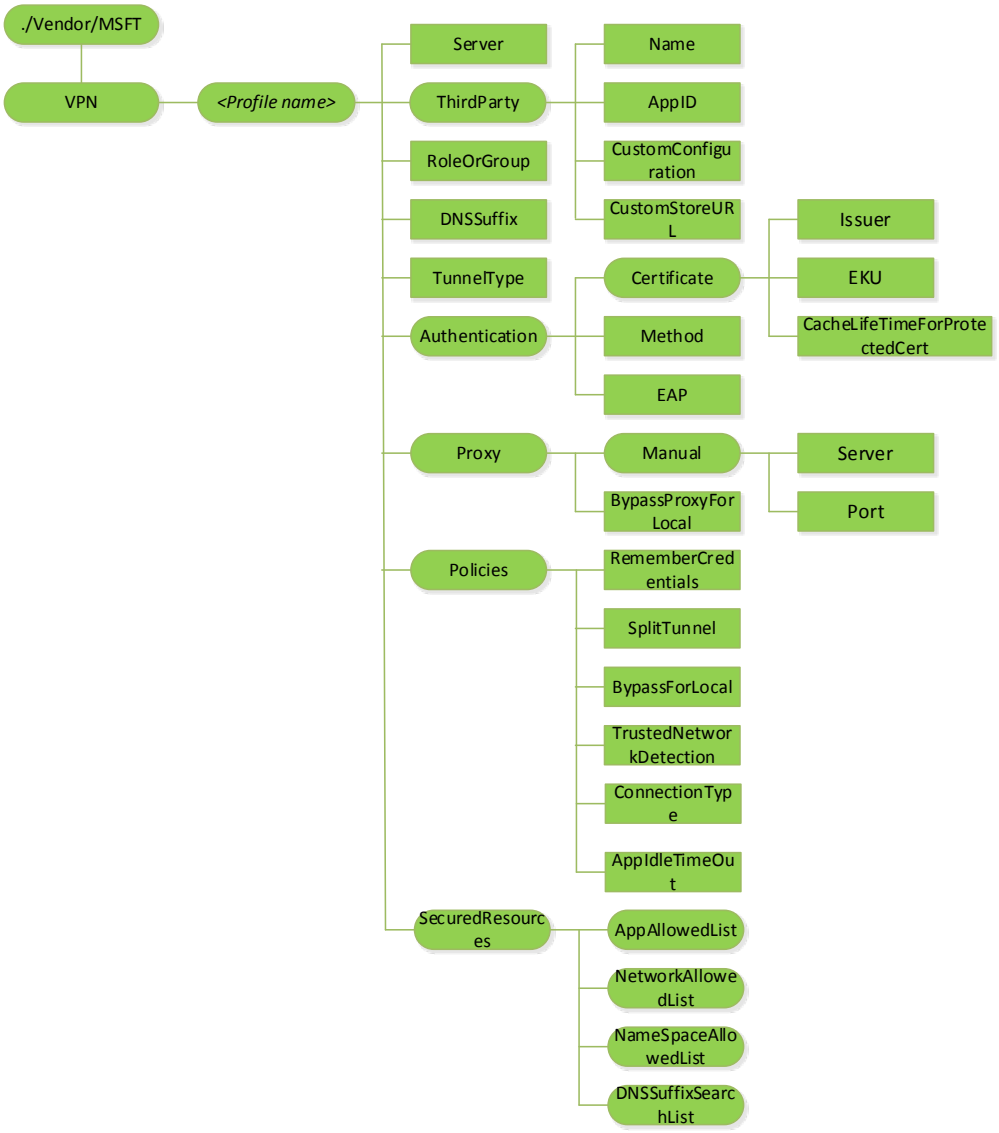
Note 1: For VPN that requires client certificate, the server MUST enroll needed client certificate first before push down VPN profile to ensure a functional VPN profile at the device. This is particularly critical for Forced channel VPN.

Note 2: VPN configuration commands should be wrapped with Atomic DM command. Refer example in this section.

Note 3: Only one VPN profile provisioning per one OMA request is supported. Multiple VPN profiles per one OMA message request is not support.

Note 4: Name-based trigerring of VPN connections is not supported.

Note 5: You can only use the **Replace** command if the setting is already configured. For first time configurations, use the **Add** command.


./Vendor/MSFT
VPN
<Profile name>
Server
ThirdParty
RoleOrGroup
DNSSuffix
TunnelType
Authentication
Proxy
Policies
SecuredResources
Name
AppID
CustomConfiguration
CustomStoreURL
Certificate
Method
EAP
Issuer
EKU
CacheLifeTimeForProtectedCert
Manual
BypassProxyForLocal
Server
Port
RememberCredentials
SplitTunnel
BypassForLocal
TrustedNetworkDetection
ConnectionType
AppIdleTimeOut
AppAllowedList
NetworkAllowedList
NameSpaceAllowedList
DNSSuffixSearchList

./Vendor/MSFT/VPN

Configuration Service Provider Root node.

./Vendor/MSFT/VPN/*

Unique alpha numeric Identifier for the profile. Supported operations include Get, Add, Replace, and Delete.

Note that profile name must not include forward slash "/".

SERVER

Required node. Public (routable) IP address or DNS name for the VPN gateway server farm. It can point to the external IP of a gateway a virtual IP for a server farm.

Type: chr

Supported Operations: Get, Add and Replace.

Examples : 208.23.45.130 or vpn.contoso.com.

TUNNELTYPE

Optional node, but required if deploying an IKEv2 VPN profile. Only a value of IKEv2 or L2TP is supported for this release.

Type: chr

Supported Operations: Get and Add.

THIRDPARTY

Optional node, but required if deploying a 3rd party SSL-VPN plugin profile. Defines a group of settings applied to SSL-VPN profile provisioning. Supported operations are: Get and Add.

THIRDPARTY/NAME

Required node if THIRDPARTY is defined for SSL-VPN profile provisioning. Supported operations are: Get and Add. Valid values are:

- JunOS Pulse
- SonicWall Mobile Connect
- F5 Big-IP Edge Client
- Checkpoint Mobile VPN

THIRDPARTY/APPID

Optional node, but required if enterprise is pushing 3rd party SSL-VPN plugin app from the private enterprise storefront. This would be a ProductID associated with the store application. The client will use this ProductID to ensure that only the "enterprise approved" plugin is initialized. Supported operations are Get, Add, Replace, and Delete. This is a String type node.

THIRDPARTY/CUSTOMCONFIGURATION

Optional node. This is an XML blob for SSL-VPN plugin specific configuration that's pushed to the device to make available for SSL-VPN plugins. Supported operations are Get, Add, Replace, and Delete. This is XML format of type CHAR.

THIRDPARTY/CustomStoreURL

Optional node, but required if enterprise is pushing 3rd party SSL-VPN plugin app from the private enterprise storefront. This node specifies 3rd party SSL-VPN plugin app's store URL link. Supported operations are Get, Add, Replace, and Delete. This is a String type node.

AUTHENTICATION

Optional node for ThirdParty VPN profiles, but required for IKEv2. A collection of configuration objects to ensure that the correct authentication policy is used on the device based on the chosen TunnelType. Supported operations are Get and Add.

AUTHENTICATION/CERTIFICATE

Optional node. A collection of nodes that enables simpler authentication experiences for end users when using VPN. Supported operations are Get and Add. This and its subnodes should not be used for IKEv2 profiles.

AUTHENTICATION/CERTIFICATE/ISSUER

Optional node. This will be of type String, and will be used to filter out the installed certificates with private keys stored in registry or TPM. This can be used in conjunction with EKU for more granular filtering. Supported operations are Get, Add, Delete, and Replace.

AUTHENTICATION/CERTIFICATE/EKU

Optional node. This Extended Key Usage node is of type String and will be used to filter out the installed certificates with private keys stored in registry or TPM. This can be used in conjunction with ISSUER for more granular filtering. Supported operations are Get, Add, Delete, and Replace.

AUTHENTICATION/EAP

Required node if IKEv2 is selected. This will define the EAP blob to be used for IKEv2 authentication. It could be either EAP-MSCHAPv2 or EAP-TLS. EAP blob is HTML encoded XML as defined in EAP Host Config schemas. You can find the schemas on msdn at http://msdn.microsoft.com/en-us/library/cc233018.aspx and http://msdn.microsoft.com/en-us/library/cc233016.aspx

Type: chr

Supported Operations: Get, Add and Replace.

AUTHENTICATION/METHOD

Required node for IKEv2 profiles. Not used for ThirdParty profiles. This specifies the authentication provider to use for VPN client authentication.

Only EAP method is supported for IKEv2 profiles. Note that for EAP, use AUTHENTICATION/EAP instead.

Type: chr

Supported Operations: Get and Add.

PROXY

Optional node. A collection of configuration objects to enable a “post-connect” proxy support for VPN. The proxy defined for this profile will be applied when this profile is active and connected.

PROXY/MANUAL

Optional node. A collection of configuration objects to enable a manual proxy with required server and port details.

PROXY/MANUAL/SERVER

Optional node. This should be set together with PORT. Its value proxy server address as a fully qualified hostname or an IP address.

Type: chr

Supported Operations: Get, Add, Replace and Delete.

Example: proxy.contoso.com

PROXY/MANUAL/PORT

Optional node. This should be set together with SERVER . Its value is the proxy server port number in the range of 1-65535.

Type: int

Supported Operations: Get, Add, Replace and Delete.

Example: 8080

PROXY/BYPASSPROXYFORLOCAL

Optional node. When this setting is enabled, any web requests to resources in the "intranet" zone will not be sent to the proxy. When this is false, the setting should be disabled and all requests should go to the proxy. When this is true, the setting is enabled and intranet requests will not go to the proxy.

Type: bool

Supported Operations: Get, Add, Replace and Delete.

Default Value: false

Example: true

SECUREDRESOURCES

A collection of configuration objects that will define the inclusion and exclusion resource lists for what should be secured over VPN. Allowed lists are applied only when POLICIES/SPLITTUNNEL node is set to true. VPN Exclusions are applied only when POLICIES/SPLITTUNNEL node is set to false.

SECUREDRESOURCES/APPALLOWEDLIST

Optional node. This will be one or many PackageFamilyNames for Enterprise LoB applications built for Windows Phone. When defined, all traffic sourced from defined apps will be secured over VPN (assuming protected networks defined allows access). They will not be able to connect directly bypassing the VPN connection. When the profile is auto-triggered, VPN will get triggered automatically by these apps.

Type: chr

Supported Operations: Get, Add, Replace and Delete.

Example: `{F05DC613-E223-40AD-ABA9-CCCE04277CD9}`

`Example: ContosoCorp.ContosoApp_jlsnulm3s397u`

SECUREDRESOURCES/NETWORKALLOWEDLIST

Optional node, but required when POLICIES/SPLITTUNNEL is set to true for IKEv2 profile. This will be one or many IP ranges defined such that all traffic to these IP ranges will be secured over VPN. Applications connecting to "protected resources" that match this list will be secured over VPN. Otherwise, they'll continue to connect directly. IP ranges are defined in the format: 10.0.0.0/8.

Type: chr

Supported Operations: Get, Add, Replace and Delete.

Example: `172.31.0.0/16`

SECUREDRESOURCES/NAMESPACEALLOWEDLIST

Optional node. This will be one or many namespaces defined such that all requests to the configured namespaces will be secured over VPN. Applications connecting to namespaces defined will be secured over VPN. Otherwise, they'll continue to connect directly. Namespaces are defined in the format: *.corp.contoso.com. Restrictions such as * or *.* or *.com.* are not allowed.

NETWORKALLOWEDLIST is still required for IKEv2 profiles for routing the traffic correctly over split tunnel.

Type: chr

Supported Operations: Get, Add, Replace and Delete.

Example: `*.corp.contoso.com`

SECUREDRESOURCES/DNSSUFFIXSEARCHLIST

Optional node. This will be one or many DNS suffixes that will be appended to shortname URLs for DNS resolution and connectivity.

Type: chr

Supported Operations: Get, Add, Replace and Delete.

Example: `corp.contoso.com`

POLICIES

Optional node. A collection of configuration objects to enforce profile-specific restrictions.

POLICIES/SPLITTUNNEL

Optional node. When this is false, all traffic goes to VPN gateway in force tunnel mode. When this is true, only specific traffic to defined "secured resources" will go to VPN gateway.

Type: bool

Supported Operations: Get, Add, Replace and Delete.

Default value: true

Example: `true`

POLICIES/REMEMBERCREDENTIALS

Optional node. When this is true, VPN traffic will remember the user credentials and provide a sign on experience.

Type: bool

Supported Operations: Get, Add, Replace and Delete.

Default value: true

Example: `true`

POLICIES/BYPASSFORLOCAL

Optional node. When this setting is set to true, requests to local resources that are available on the same Wifi network as the VPN client will bypass the VPN. For example, if enterprise policy for VPN requires force tunnel for VPN, but enterprise intends to allow the remote user to connect locally to media center in their home, then this option should be set to true. The user will be able to bypass VPN for local subnet traffic. When this is set to false, the setting should be disabled and no subnet exceptions are allowed.

Type: bool

Supported Operations: Get, Add, Replace and Delete.

Default value: false

Example: `true`

POLICIES/TRUSTEDNETWORKDETECTION

Optional node. When this setting is set to true, VPN will not connect when the user is on their corporate wireless network where protected resources are directly accessible to the device. When this is set to false, VPN will connect over corporate wireless network. This node has a dependency on node DNSSuffix to be set in order to detect the corporate wireless network.

Type: bool

Supported Operations: Get, Add, Replace and Delete.

Default value: false

Example: `true`

POLICIES/CONNECTIONTYPE

Optional node. Valid values are:

- Triggering: VPN automatically connects as applications require connectivity to protected resource. Life cycle of VPN is based on applications using the VPN. Recommended setting for optimizing usage of power resources.
- Manual: User must manually connect / disconnect VPN.
- (new for GDR2) AlwaysOn: VPN is always connected when there is network connection.

Type: chr

Supported Operations: Get, Add, Replace.

Default value: Triggering

Example: `Manual`

POLICIES/APPIDLETIMEOUT (new for GDR2)

Optional node. For use in VPN with triggering connection type. The connection manager automatically disconnects the VPN if the application is idle with not connectivity request. The idle time out setting ranges from 30 seconds to 86400 seconds (24 hours), with a default value of 600 seconds (10 minutes)

Type: int

Supported Operations: Get, Add, Replace.

Default value: 600 (10 min)

Example: `30`

DNSSUFFIX

Optional node. Required setting to push down the primary connection specific DNS suffix.

Type: chr

Supported Operations: Get, Add, Replace and Delete.

Example: `corp.contoso.com`

Examples

IKEv2 VPN profile using EAP-TLS as authentication method with Server certificate validation turned on.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Atomic>
      <CmdID>8000</CmdID>
      <Add>
        <CmdID>8001</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Server</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>vpntestgateway.vpntest.com</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8002</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/TunnelType</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>IKEv2</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8004</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Authentication/Method</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>EAP</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8005</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Authentication/EAP</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>
            &lt;EapHostConfig
xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;&gt;
            &lt;EapMethod&gt;
            &lt;Type
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;13&lt;/Type&gt;
```

```
            &lt;VendorId
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/VendorId&gt;
            &lt;VendorType
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/VendorType&gt;
            &lt;AuthorId
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/AuthorId&gt;
            &lt;/EapMethod&gt;
            &lt;Config
xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;&gt;
            &lt;Eap
xmlns=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;&gt;
            &lt;Type&gt;13&lt;/Type&gt;
            &lt;EapType
xmlns=&quot;http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV1&quot;&gt;
            &lt;CredentialsSource&gt;
            &lt;CertificateStore&gt;
            &lt;SimpleCertSelection&gt;true&lt;/SimpleCertSelection&gt;
            &lt;/CertificateStore&gt;
            &lt;/CredentialsSource&gt;
            &lt;ServerValidation&gt;

&lt;DisableUserPromptForServerValidation&gt;false&lt;/DisableUserPromptForServerValidation&gt;
            &lt;ServerNames&gt;&lt;/ServerNames&gt;
            &lt;TrustedRootCA&gt;50 d4 d5 0e 9c f5 0e 9e 17 34 2c 83 79 11 ed 21 39 52 bf
f3&lt;/TrustedRootCA&gt;
            &lt;/ServerValidation&gt;
            &lt;DifferentUsername&gt;false&lt;/DifferentUsername&gt;
            &lt;PerformServerValidation
xmlns=&quot;http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2&quot;&gt;true&l
t;/PerformServerValidation&gt;
            &lt;AcceptServerName
xmlns=&quot;http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2&quot;&gt;false&
lt;/AcceptServerName&gt;
            &lt;/EapType&gt;
            &lt;/Eap&gt;
            &lt;/Config&gt;
            &lt;/EapHostConfig&gt;
          </Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8006</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Policies/ConnectionType</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>Manual</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8007</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/VPN/EapTls/SecuredResources/NetworkAllowedList/Networks000</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
```

```
          <Data>192.168.0.0/16</Data>
        </Item>
      </Add>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

VPN profile using EAP-TLS authentication method

```
<SyncML xmlns="SYNCML:SYNCML1.2">
<SyncBody>
    <Atomic>
      <CmdID>8000</CmdID>
      <Add>
        <CmdID>8001</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Server</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>wp.test.com</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8002</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/TunnelType</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>IKEv2</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8004</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Authentication/Method</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>EAP</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8005</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/Authentication/EAP</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>
            &lt;EapHostConfig
            xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;
```

```
            xmlns:eapCommon=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;

xmlns:baseEap=&quot;http://www.microsoft.com/provisioning/BaseEapMethodConfig&quot;&gt;
            &lt;EapMethod&gt;
            &lt;eapCommon:Type&gt;13&lt;/eapCommon:Type&gt;
            &lt;eapCommon:AuthorId&gt;0&lt;/eapCommon:AuthorId&gt;
            &lt;/EapMethod&gt;
            &lt;Config

xmlns:baseEap=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;

xmlns:eapTls=&quot;http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV1&quot;&gt
;
            &lt;baseEap:Eap&gt;
            &lt;baseEap:Type&gt;13&lt;/baseEap:Type&gt;
            &lt;eapTls:EapType&gt;
            &lt;eapTls:CredentialsSource&gt;
            &lt;eapTls:CertificateStore&gt;
            &lt;eapTls:SimpleCertSelection&gt;true&lt;/eapTls:SimpleCertSelection&gt;
            &lt;/eapTls:CertificateStore&gt;
            &lt;/eapTls:CredentialsSource&gt;
            &lt;/eapTls:EapType&gt;
            &lt;/baseEap:Eap&gt;
            &lt;/Config&gt;
            &lt;/EapHostConfig&gt;
          </Data>
        </Item>
      </Add>
      <Add>
        <!-- Network trigger: 1.2.3.4/16 -->
        <CmdID>8008</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/VPN/EapTls/SecuredResources/NetworkAllowedList/Networks000</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>1.2.3.4/16</Data>
        </Item>
      </Add>
      <Add>
        <!-- Host trigger: + -->
        <CmdID>8023</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/VPN/EapTls/SecuredResources/NameSpaceAllowedList/NameSpace001</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>+</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8024</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapTls/DNSSuffix</LocURI>
          </Target>
```

```
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>corp.test.com</Data>
        </Item>
      </Add>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

VPN profile using EAP-MSCHAPV2 authentication method.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
<SyncBody>
    <Atomic>
      <CmdID>8000</CmdID>
      <Add>
        <CmdID>8001</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapMsChapv2/Server</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>wp.test.com</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8002</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapMsChapv2/TunnelType</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>IKEv2</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8004</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapMsChapv2/Authentication/Method</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>EAP</Data>
        </Item>
      </Add>
      <Add>
        <CmdID>8005</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapMsChapv2/Authentication/EAP</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
```

```
          <Data>
            &lt;EapHostConfig
            xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;
            xmlns:eapCommon=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;

xmlns:baseEap=&quot;http://www.microsoft.com/provisioning/BaseEapMethodConfig&quot;

xmlns:msChapV2=&quot;http://www.microsoft.com/provisioning/MsChapV2ConnectionPropertiesV1&quot
;&gt;
            &lt;EapMethod&gt;
            &lt;eapCommon:Type&gt;26&lt;/eapCommon:Type&gt;
            &lt;eapCommon:AuthorId&gt;0&lt;/eapCommon:AuthorId&gt;
            &lt;/EapMethod&gt;
            &lt;Config

xmlns:baseEap=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;

xmlns:msPeap=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV1&quot;

xmlns:msChapV2=&quot;http://www.microsoft.com/provisioning/MsChapV2ConnectionPropertiesV1&quot
;&gt;
            &lt;baseEap:Eap&gt;
            &lt;baseEap:Type&gt;26&lt;/baseEap:Type&gt;
            &lt;msChapV2:EapType&gt;

&lt;msChapV2:UseWinLogonCredentials&gt;false&lt;/msChapV2:UseWinLogonCredentials&gt;
            &lt;/msChapV2:EapType&gt;
            &lt;/baseEap:Eap&gt;
            &lt;/Config&gt;
            &lt;/EapHostConfig&gt;
          </Data>
        </Item>
      </Add>
      <Add>
        <!-- Network trigger: 1.2.3.4/16 -->
        <CmdID>8008</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/VPN/EapMsChapv2/SecuredResources/NetworkAllowedList/Networks000</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>1.2.3.4/16</Data>
        </Item>
      </Add>
      <Add>
        <!-- Host trigger: *.corp.test.com -->
        <CmdID>8022</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/VPN/EapMsChapv2/SecuredResources/NameSpaceAllowedList/NameSpace000</LocU
RI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>*.corp.test.com</Data>
        </Item>
      </Add>
```

```
      <Add>
        <CmdID>8024</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/VPN/EapMsChapv2/DNSSuffix</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>corp.test.com</Data>
        </Item>
      </Add>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

VPN single sign on configuration

For all supported VPN profiles (i.e. IKEv2, 3rd party SSL-VPN plugins), Windows Phone 8.1 supports a single sign-on user experience where users will be authenticated to all NTLM-protected domain resources inside an enterprise when VPN is connected for resources that are protected by the VPN profile. The user only needs to enter username/password once in the authentication dialog (if VPN gateway authentication is U/P based), or in the VPN profile edit page (found in Settings → VPN) if the VPN gateway authentication is certificate based. The intranet sites accessed by the user via IE or other Line of Business applications (with a specified capability) will not ask user for their username/password after VPN gets connected.

To achieve this function, the MDM server needs to configure the device's IE intranet zone settings. This will enable Internet Explorer to treat certain "intranet sites" as trusted, and will provide a single sign-on experience. The MDM server should configure intranet zone settings (URLs, domains, IPs) to following reg key path via Registry CSP:

HKCU/Software/Microsoft/Windows/CurrentVersion/Internet Settings/ZoneMap/Domains

NOTE 1: MDM sever could configure multiple nodes under HKCU/Software/Microsoft/Windows/CurrentVersion/Internet Settings/ZoneMap/Domains. Refer FQDN part in kb article http://support.microsoft.com/kb/303650 for detailed description on intranet settings configuration.
NOTE 2: using registry key to configure intranet zone is a temporarily solution and subject to be replaced with a new structured OMA URL path in future release.

The following example configures any URLs under *.internal.contoso.com to be intranet URL that IE could leverage via VPN channel.

```
<Atomic>
      <CmdID>8</CmdID>
      <Add>
        <CmdID>8001</CmdID>
        <Item>
          <Target>
            <LocURI>
./Vendor/MSFT/Registry/HKCU/Software/Microsoft/Windows/CurrentVersion/Internet%20Settings/Zone
Map/Domains/contoso.com/*.internal/*
</LocURI>
```

```
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">int</Format>
          </Meta>
          <Data>1</Data>
        </Item>
      </Add>
<Atomic>
```

WiFi configuration service provider (New in Windows Phone 8.1)

The Wi-Fi configuration service provider (CSP) provides functionality to add or delete Wi-Fi networks on a Windows Phone device. The CSP accepts a SyncML input and converts it to a network profile that is installed on the device. This profile enables the phone to connect to the Wi-Fi network when it is in range.

If the authentication method needs a certificate (e.g. EAP-TLS requires client certificates), this must be configured through the certificate store CSP first. The WiFi CSP does not provide that functionality; instead the Wi-Fi profile can specify characteristics of the certificate to be used for choosing the right certificate for that network. And the server should successfully enroll needed client certificate first before push down WiFi network configuration

Note 1: Since Windows Phone Emulators do not have Wi-Fi radio support, Wi-Fi network configuration cannot be tested end-to-end with an emulator. A Wi-Fi network can still be provisioned using the WiFi CSP and the network should be visible in the Wi-Fi Settings page, but connectivity to that network cannot be tested.

Note 2: For WEP, WPA, and WPA2-based networks, the passkey must be included in the network configuration in plaintext. It will be encrypted automatically while storing on the device.

Note 3: WlanXml blob is sent in OMA SyncML XML message as chr. The profile XML content needs to be XML escaped in OMA message.

Note 4: keyMaterial if exists in the wlanxml blob needs to come after keyType and protected elements like documented in MSDN - http://msdn.microsoft.com/en-us/library/windows/desktop/aa370032(v=vs.85).aspx

Note 5: For EAP-TLS profile, the server must successfully configure and enroll the required client certificate first before push down WiFi profile.

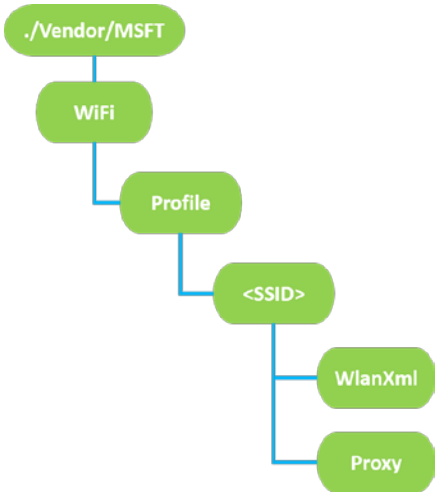
Note 6: Self signed server certificate works for EAP-TLS/PEAP-MSCHAPv2 but it isn't supported in EAP-TTLS.

Note 7: The SSID of the Wi-Fi networks part of the LocURI node, which must be a valid URI based on RFC 2396. This requires that all non-ASCII characters must be escaped using a %-character. Unicode characters without the necessary escaping are not supported.

NOTE 8: The <*name*>*name_goes_here*</*name*><*SSIDConfig*> must match <*SSID*><*name*> *name_goes_here*</*name*></*SSID*>

NOTE 9: Windows Phone does not support EapHostUserCredentials for Enterprise Wi-Fi WlanXml blob.

The following diagram shows the Wi-Fi configuration service provider in tree format.



Profile
Each Wi-Fi network configuration is represented by a profile object. This network profile includes all the information required for the phone to connect to that network – for example, the SSID, authentication and encryption methods and passphrase in case of WEP or WPA2 networks. Supported operation: Get.

<SSID>
The SSID of the Wi-Fi network (maximum length 32 bytes, case-sensitive). This can be represented in ASCII. Supported operations: Get. SSID is added when WlanXML node is added, and deleted when WlanXml is deleted.

WlanXml
This is the XML describing the network configuration and follows the Windows WLAN_profile Schema (MSDN documentation). Supported operations: Get, Add, Delete, Replace

Proxy
A proxy server host and port can be specified per connection for Windows Phone. The format is **host:port**, where host can be one of the following:

- a registered host name, such as server name, FQDN, or Single Label Name (SLN), such as myweb instead of myweb.contoso.com.
- IPv4 address
- IPv6/IPvFuture address

If it is an IPvFuture address, then it must be specified as an IP literal as "[" (IP v6 address / IPvFuture) "]", such as "[2441:4880:28:3:204:76ff:f43f:6eb]:8080".

Supported operations: Get, Add, Delete, Replace

Best Practices

NOTE: The <name>*name_goes_here*</name><SSIDConfig> must match the <SSID><name>*name_goes_here*</name></SSID>

Examples

Adding an open network with SSID 'MyNetwork' and no proxy

```
<Atomic>
      <CmdID>300</CmdID>
      <Add>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MyNetwork
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MyNetwork&lt;/name&gt;&lt;/SSID&gt;&lt;
/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/connectionType&gt;&lt;connectionMode&gt;manual&lt
;/connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&lt;authEncryption&gt;&lt;authentication&gt;ope
n&lt;/authentication&gt;&lt;encryption&gt;none&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;/s
ecurity&gt;&lt;/MSM&gt;&lt;/WLANProfile&gt; </Data>
        </Item>
      </Add>
</Atomic>
```

Removing a network with SSID 'MyNetwork' and no proxy

```
<Atomic>
      <CmdID>300</CmdID>
      <Delete>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
        </Item>
      </Delete>
</Atomic>
```

Note: Deletion for all authentication types of networks is the same.Querying WiFi profiles SSID installed MDM server

```
<Get>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile</LocURI>
          </Target>
        </Item>
</Get>
```

Response from the phone (two SSID returned)

```
<Results>
       <CmdID>3</CmdID>
       <MsgRef>1</MsgRef>
       <CmdRef>301</CmdRef>
       <Item>
              <Source><LocURI>./Vendor/MSFT/WiFi/Profile</LocURI></Source>
              <Meta><Format xmlns="syncml:metinf">node</Format></Meta>
              <Data>TestWLAN1/TestWLAN2</Data>
       </Item>
</Results>
```

Adding an open network with SSID 'My Network' and a proxy with url 'test' and port '80'

```
<Atomic>
      <CmdID>300</CmdID>
      <Add>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MyNetwork
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MyNetwork&lt;/name&gt;&lt;/SSID&gt;&lt;
/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/connectionType&gt;&lt;connectionMode&gt;manual&lt
;/connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&lt;authEncryption&gt;&lt;authentication&gt;ope
n&lt;/authentication&gt;&lt;encryption&gt;none&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;/s
ecurity&gt;&lt;/MSM&gt;&lt;/WLANProfile&gt; </Data>
        </Item>
      </Add>
      <Add>
        <CmdID>302</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/Proxy</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>test:80</Data>
        </Item>
      </Add>
</Atomic>
```

Note: Addition of proxy to networks with any other authentication type also works in the same way as this example.

Adding a WPA2 PSK network with SSID 'My Network' and no proxy

```
<Atomic>
      <CmdID>300</CmdID>
      <Add>
        <CmdID>301</CmdID>
        <Item>
          <Target>
```

```
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MyNetwork
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MyNetwork&lt;/name&gt;&lt;/SSID&gt;&lt;
/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/connectionType&gt;&lt;connectionMode&gt;manual&lt
;/connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&lt;authEncryption&gt;&lt;authentication&gt;WPA
2PSK&lt;/authentication&gt;&lt;encryption&gt;AES&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;
sharedKey&gt;&lt;keyType&gt;passPhrase&lt;/keyType&gt;&lt;protected&gt;false&lt;/protected&gt;
&lt;keyMaterial&gt;123456789&lt;/keyMaterial&gt;&lt;/sharedKey&gt;&lt;/security&gt;&lt;/MSM&gt
;&lt;/WLANProfile&gt; </Data>
        </Item>
      </Add>
</Atomic>
```

Adding WPA PSK network with SSID 'MyNetwork' and no proxy

```
<Atomic>
      <CmdID>300</CmdID>
      <Add>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MyNetwork
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MyNetwork&lt;/name&gt;&lt;/SSID&gt;&lt;
/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/connectionType&gt;&lt;connectionMode&gt;manual&lt
;/connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&lt;authEncryption&gt;&lt;authentication&gt;WPA
PSK&lt;/authentication&gt;&lt;encryption&gt;TKIP&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;
sharedKey&gt;&lt;keyType&gt;passPhrase&lt;/keyType&gt;&lt;protected&gt;false&lt;/protected&gt;
&lt;keyMaterial&gt;123456789&lt;/keyMaterial&gt;&lt;/sharedKey&gt;&lt;/security&gt;&lt;/MSM&gt
;&lt;/WLANProfile&gt; </Data>
        </Item>
      </Add>
</Atomic>
```

Adding PEAP-MSCHAPv2 network with SSID 'MyNetwork' and no proxy. (Default is to prompt the user for server certificate validation)

```
<Atomic>
      <CmdID>300</CmdID>
      <Add>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
```

```
          <Data>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MyNetwork
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MyNetwork&lt;/name&gt;&lt;/SSID&gt;&lt;
nonBroadcast&gt;false&lt;/nonBroadcast&gt;&lt;/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/con
nectionType&gt;&lt;connectionMode&gt;manual&lt;/connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&
lt;authEncryption&gt;&lt;authentication&gt;WPA2&lt;/authentication&gt;&lt;encryption&gt;AES&lt
;/encryption&gt;&lt;useOneX&gt;true&lt;/useOneX&gt;&lt;/authEncryption&gt;&lt;OneX
xmlns=&quot;http://www.microsoft.com/networking/OneX/v1&quot;&gt;&lt;authMode&gt;user&lt;/auth
Mode&gt;&lt;EAPConfig&gt;&lt;EapHostConfig
xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;&gt;&lt;EapMethod&gt;&lt;
Type
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;25&lt;/Type&gt;&lt;Vendor
Id
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/VendorId&gt;&lt;Ven
dorType
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/VendorType&gt;&lt;A
uthorId
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/AuthorId&gt;&lt;/Ea
pMethod&gt;&lt;Config
xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;&gt;&lt;Eap
xmlns=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;&gt;&lt;T
ype&gt;25&lt;/Type&gt;&lt;EapType
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV1&quot;&gt;&lt;Se
rverValidation&gt;&lt;DisableUserPromptForServerValidation&gt;false&lt;/DisableUserPromptForSe
rverValidation&gt;&lt;ServerNames&gt;&lt;/ServerNames&gt;&lt;TrustedRootCA&gt;InsertCertThumbP
rintHere&lt;/TrustedRootCA&gt;&lt;/ServerValidation&gt;&lt;FastReconnect&gt;true&lt;/FastRecon
nect&gt;&lt;InnerEapOptional&gt;false&lt;/InnerEapOptional&gt;&lt;Eap
xmlns=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;&gt;&lt;T
ype&gt;26&lt;/Type&gt;&lt;EapType
xmlns=&quot;http://www.microsoft.com/provisioning/MsChapV2ConnectionPropertiesV1&quot;&gt;&lt;
UseWinLogonCredentials&gt;false&lt;/UseWinLogonCredentials&gt;&lt;/EapType&gt;&lt;/Eap&gt;&lt;
EnableQuarantineChecks&gt;false&lt;/EnableQuarantineChecks&gt;&lt;RequireCryptoBinding&gt;fals
e&lt;/RequireCryptoBinding&gt;&lt;PeapExtensions&gt;&lt;PerformServerValidation
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2&quot;&gt;true&l
t;/PerformServerValidation&gt;&lt;AcceptServerName
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2&quot;&gt;false&
lt;/AcceptServerName&gt;&lt;PeapExtensionsV2
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2&quot;&gt;&lt;Al
lowPromptingWhenServerCANotFound
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV3&quot;&gt;true&l
t;/AllowPromptingWhenServerCANotFound&gt;&lt;/PeapExtensionsV2&gt;&lt;/PeapExtensions&gt;&lt;/
EapType&gt;&lt;/Eap&gt;&lt;/Config&gt;&lt;/EapHostConfig&gt;&lt;/EAPConfig&gt;&lt;/OneX&gt;&lt
;/security&gt;&lt;/MSM&gt;&lt;/WLANProfile&gt; </Data>
        </Item>
      </Add>
</Atomic>
```

Adding PEAP-MSCHAPv2 network with SSID 'MyNetwork' and root CA validation for server certificate

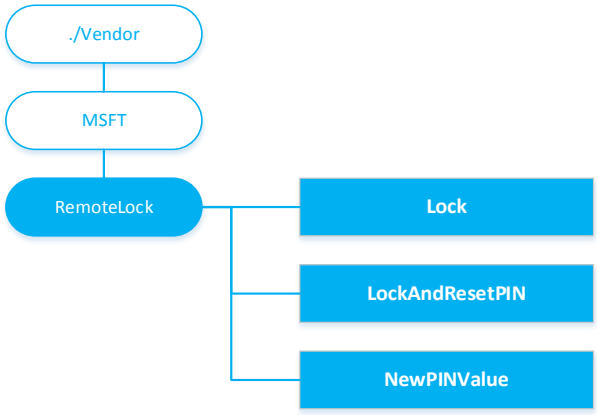
```
<Atomic>
      <CmdID>300</CmdID>
      <Add>
        <CmdID>301</CmdID>
        <Item>
          <Target>
            <LocURI>./Vendor/MSFT/WiFi/Profile/MyNetwork/WlanXml</LocURI>
          </Target>
          <Meta>
```

```
            <Format xmlns="syncml:metinf">chr</Format>
          </Meta>
          <Data>&lt;?xml version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;MyNetwork
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;MyNetwork&lt;/name&gt;&lt;/SSID&gt;&lt;
nonBroadcast&gt;false&lt;/nonBroadcast&gt;&lt;/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/con
nectionType&gt;&lt;connectionMode&gt;manual&lt;/connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&
lt;authEncryption&gt;&lt;authentication&gt;WPA2&lt;/authentication&gt;&lt;encryption&gt;AES&lt
;/encryption&gt;&lt;useOneX&gt;true&lt;/useOneX&gt;&lt;/authEncryption&gt;&lt;OneX
xmlns=&quot;http://www.microsoft.com/networking/OneX/v1&quot;&gt;&lt;authMode&gt;user&lt;/auth
Mode&gt;&lt;EAPConfig&gt;&lt;EapHostConfig
xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;&gt;&lt;EapMethod&gt;&lt;
Type
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;25&lt;/Type&gt;&lt;Vendor
Id
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/VendorId&gt;&lt;Ven
dorType
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/VendorType&gt;&lt;A
uthorId
xmlns=&quot;http://www.microsoft.com/provisioning/EapCommon&quot;&gt;0&lt;/AuthorId&gt;&lt;/Ea
pMethod&gt;&lt;Config
xmlns=&quot;http://www.microsoft.com/provisioning/EapHostConfig&quot;&gt;&lt;Eap
xmlns=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;&gt;&lt;T
ype&gt;25&lt;/Type&gt;&lt;EapType
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV1&quot;&gt;&lt;Se
rverValidation&gt;&lt;DisableUserPromptForServerValidation&gt;true&lt;/DisableUserPromptForSer
verValidation&gt;&lt;ServerNames&gt;&lt;/ServerNames&gt;&lt;TrustedRootCA&gt;
InsertCertThumbPrintHere
&lt;/TrustedRootCA&gt;&lt;/ServerValidation&gt;&lt;FastReconnect&gt;true&lt;/FastReconnect&gt;
&lt;InnerEapOptional&gt;false&lt;/InnerEapOptional&gt;&lt;Eap
xmlns=&quot;http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1&quot;&gt;&lt;T
ype&gt;26&lt;/Type&gt;&lt;EapType
xmlns=&quot;http://www.microsoft.com/provisioning/MsChapV2ConnectionPropertiesV1&quot;&gt;&lt;
UseWinLogonCredentials&gt;false&lt;/UseWinLogonCredentials&gt;&lt;/EapType&gt;&lt;/Eap&gt;&lt;
EnableQuarantineChecks&gt;false&lt;/EnableQuarantineChecks&gt;&lt;RequireCryptoBinding&gt;fals
e&lt;/RequireCryptoBinding&gt;&lt;PeapExtensions&gt;&lt;PerformServerValidation
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2&quot;&gt;true&l
t;/PerformServerValidation&gt;&lt;AcceptServerName
xmlns=&quot;http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2&quot;&gt;false&
lt;/AcceptServerName&gt;&lt;/PeapExtensions&gt;&lt;/EapType&gt;&lt;/Eap&gt;&lt;/Config&gt;&lt;
/EapHostConfig&gt;&lt;/EAPConfig&gt;&lt;/OneX&gt;&lt;/security&gt;&lt;/MSM&gt;&lt;/WLANProfile
&gt; </Data>
        </Item>
      </Add>
</Atomic>
```

RemoteLock configuration service provider (New in Windows Phone 8.1)

The RemoteLock CSP supports the ability to lock a device that has a PIN set on the device or reset the PIN on a device that may or may not have a PIN set.

The following diagram shows the RemoteLock configuration service provider in a tree format.



/RemoteLock/Lock

Required. The node accepts requests to lock the device screen. The device screen will lock immediately if a PIN has been set. If no PIN is set, the lock request is ignored and the OMADM (405) Forbidden error is returned over the management channel. All OMADM errors are listed here in the protocol specification. The supported operation is Exec.

Status	Description	Meaning [Standard]
(200) OK	Device was successfully locked	The command and the associated Alert action are completed successfully.
(405)	Device could not be locked because there is no PIN currently set on the device	The requested command is not allowed on the target.
(500) Command failed	Device was not locked for some unknown reason	Non-specific errors were created by the recipient while attempting to complete the command.

RemoteLock/LockAndResetPIN

This node can be used to lock and reset the PIN on the device. It is used in conjunction with the NewPINValue node. After <Exec> has been called on this node and succeeds, the previous PIN will no longer work or be recoverable in any way. The supported operation is Exec.

This node will return the following status. All OMADM errors are listed here in the protocol specification.

Status	Description	Meaning
(200) OK	Device has been locked with a new password which has been reset	The command and the associated Alert action are completed successfully.
(500) Command failed	N/A	Non-specific errors created by the recipient while attempting to complete the command.

RemoteLock/NewPINValue

This node contains the PIN after Exec has been called on /RemoteLock/NewPINValue. If LockAndResetPIN has never been called, the value will be null. If Get is called on this node after a successful Exec call on /RemoteLock/NewPINValue, then the new PIN will be provided. If another Get command is called on this node, the value will be null. If the IT admin needs to reset the PIN again, then another LockAndResetPIN <Exec> can be communicated to the device to generate a new PIN. The PIN value will conform to the minimum PIN complexity requirements of the merged policies that are set on the device. If no PIN policy has been set on the device, the device will generate an 8-digit numeric PIN and set the device to have this PIN. The data type returned is a string. The supported operation is Get. A Get operation on this node must follow an Exec on the /RemoteLock/LockAndResetPIN node in the proper order and in the same syncml message. The Sequence tag can be used to guarantee the order in which commands are processed.

NOTE: It is possible, though highly unprovable, that alphanumeric PINs may contain offensive words. It is at the discretion of the IT administrator to execute another LockAndPinReset if the alphanumeric PIN does not conform to the IT administrator's company policies.

Examples

Initiate a remote lock of the device.

```
<Exec>
       <CmdID>1</CmdID>
       <Item>
              <Target>
              <LocURI>./Vendor/MSFT/RemoteLock/Lock </LocURI>
              </Target>
       </Item>
</Exec>
```

Initiate a remote lock and PIN reset of the device.

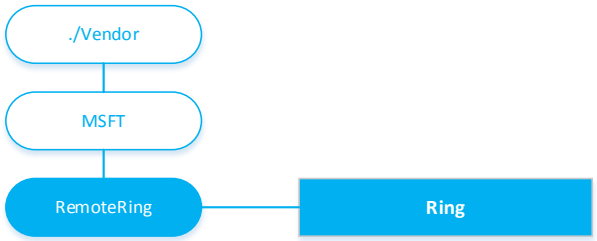
Please note that in order to retrieve the new device-generated PIN successfully, the commands must be executed together and in the proper sequence as listed below.

```
<Sequence>
    <CmdID>1</CmdID>
    <Exec>
        <CmdID>2</CmdID>
        <Item>
            <Target>
                <LocURI>./Vendor/MSFT/RemoteLock/LockAndResetPIN </LocURI>
            </Target>
        </Item>
    </Exec>
    <Get>
       <CmdID>3</CmdID>
<Item>
            <Target>
                <LocURI>./Vendor/MSFT/RemoteLock/NewPINValue </LocURI>
            </Target>
        </Item>
    </Get>
</Sequence>
```

RemoteRing configuration service provider (New in Windows Phone 8.1)

The RemoteRing CSP can be used to remotely trigger a device to produce an audible ringing sound regardless of the volume that is set on the device.

The following diagram shows the RemoteRing configuration service provider in tree format.



/RemoteRing/Ring
Required. The node accepts requests to ring the device. The supported operation is Exec.

Examples

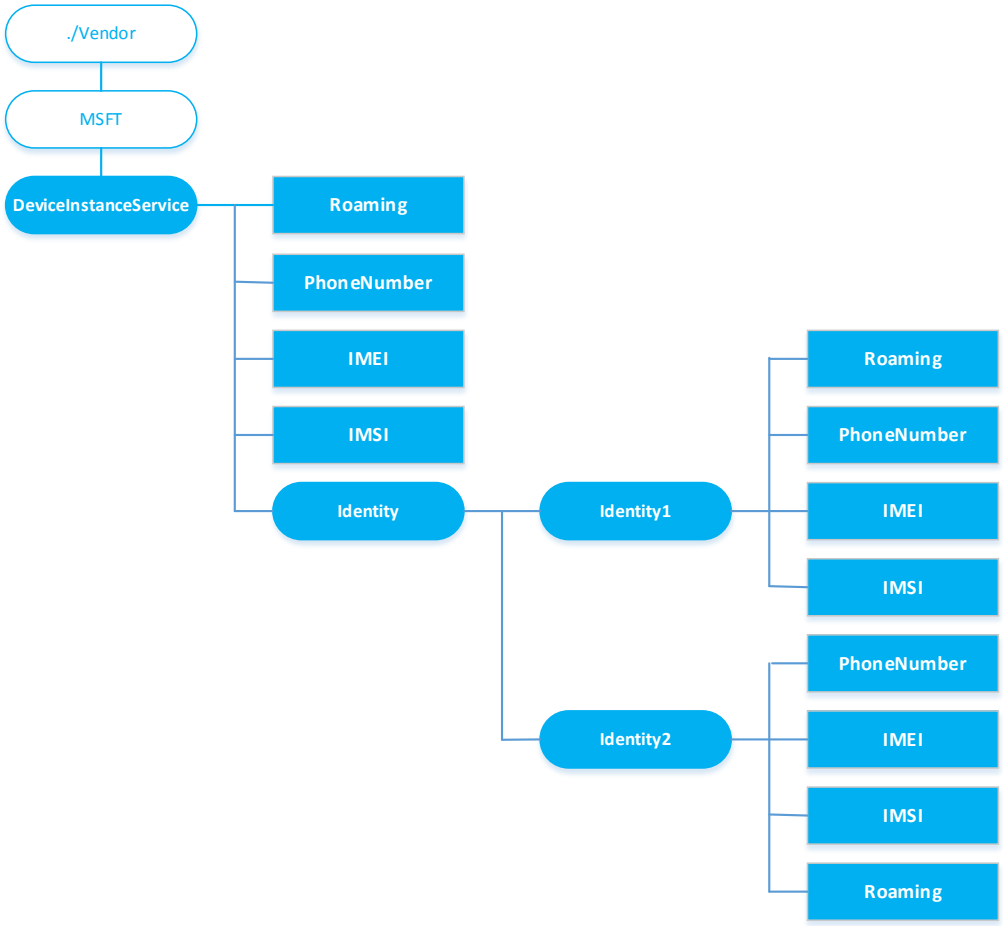
The following sample shows how to initiate a remote ring on the device.

```
<Exec>
  <CmdID>5</CmdID>
    <Item>
    <Target>
      <LocURI>./Vendor/MSFT/RemoteRing/Ring </LocURI>
    </Target>
  </Item>
</Exec>
```

DeviceInstanceService configuration service provider

DeviceInstanceService CSP provides some device inventory information that could be useful for enterprise. Additionally, this CSP supports querying two different phone numbers in the case of dual SIM. The URIs for SIM 1 and SIM 2 are ./Vendor/MSFT/DeviceInstanceService/Identity/Identity1 and ./Vendor/MSFT/DeviceInstanceService/Identity/Identity2 respectively.

The following diagram shows the DeviceInstanceService configuration service provider in tree format.



./Vendor/MSFT/**DeviceInstanceService**

- Description: DeviceInstanceService CSP root node.
- Format: node
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/**Roaming**

- Description: Present device cellular roaming status. In case of dual SIM mode when the device supports two different phone numbers, querying SIM 1 explicitly with ./Vendor/MSFT/DeviceInstanceService/Identify1/Roaming is functionally equivalent to using ./Vendor/MSFT/DeviceInstanceService/Roaming.
- Format: bool
- Supported operations: Get

- Occurrence: One
- Supported value:
    - False: device cellular is not in roaming
    - True: device cellular is in roaming

./Vendor/MSFT/DeviceInstanceService/**PhoneNumber**

- Description: Present device phone number. In case of dual SIM mode when the device supports two different phone numbers, querying SIM 1 explicitly with ./Vendor/MSFT/DeviceInstanceService/Identify1/PhoneNumber is functionally equivalent to using ./Vendor/MSFT/DeviceInstanceService/PhoneNumber.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/**IMSI**

- Description: Present first 6 digits of device IMSI number (Mobile Country Code, Mobile Network Code). In case of dual SIM mode when the device supports two different phone numbers, querying SIM 1 explicitly with ./Vendor/MSFT/DeviceInstanceService/Identify1/IMSI is functionally equivalent to using ./Vendor/MSFT/DeviceInstanceService/IMSI.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/**IMEI**

- Description: Present device IMEI number. In case of dual SIM mode when the device supports two different phone numbers, querying SIM 1 explicitly with ./Vendor/MSFT/DeviceInstanceService/Identify1/IMEI is functionally equivalent to using ./Vendor/MSFT/DeviceInstanceService/IMEI.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService**/Identity**

- Description: parent node to group per SIM specific information in case of dual SIM mode.
- Format: node
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/**Identity1**

- Description: parent node to group SIM1 specific information in case of dual SIM mode.
- Format: node
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/**Identity2**

- Description: parent node to group SIM2 specific information in case of dual SIM mode.
- Format: node

- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity1/**PhoneNumber**

- Description: Present device phone number for SIM1.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity1/**IMSI**

- Description: Present first 6 digits of device IMSI number for SIM1.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity1/**IMEI**

- Description: Present device IMEI number for SIM1.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity1/**Roaming**

- Description: Present device cellular roaming status. In case of dual SIM mode when the device supports two different phone numbers, present roaming status for SIM1.
- Format: bool
- Supported operations: Get
- Occurrence: One
- Supported value:
    - False: device cellular is not in roaming
    - True: device cellular is in roaming

./Vendor/MSFT/DeviceInstanceService/Identity/Identity2/**PhoneNumber**

- Description: Present device phone number for SIM2.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity2/**IMSI**

- Description: Present first 6 digits of device IMSI number for SIM2.
- Format: chr
- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity2/**IMEI**

- Description: Present device IMEI number for SIM2.
- Format: chr

- Supported operations: Get
- Occurrence: One

./Vendor/MSFT/DeviceInstanceService/Identity/Identity2/**Roaming**

- Description: Present device cellular roaming status. In case of dual SIM mode when the device supports two different phone numbers, present roaming status for SIM2.
- Format: bool
- Supported operations: Get
- Occurrence: One
- Supported value:
    - False: device cellular is not in roaming
    - True: device cellular is in roaming

Examples

The following sample shows how to query roaming status and phone number on the device.

```
<Get>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/DeviceInstanceService/Roaming</LocURI>
        </Target>
      </Item>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/DeviceInstanceService/PhoneNumber</LocURI>
        </Target>
      </Item>
</Get>

Response from the phone
<Results>
       <CmdID>3</CmdID>
       <MsgRef>1</MsgRef>
       <CmdRef>2</CmdRef>
       <Item>
              <Source><LocURI>./Vendor/MSFT/DeviceInstanceService/Roaming</LocURI></Source>
              <Meta><Format xmlns="syncml:metinf">bool</Format></Meta>
              <Data>false</Data>
       </Item>
       <Item>
              <Source><LocURI>./Vendor/MSFT/DeviceInstanceService/PhoneNumber</LocURI></Source
              >
              <Data>+14254458055</Data>
       </Item>
</Results>
```

EnterpriseAssignedAccess configuration service provider (New in Windows Phone 8.1)

The EnterpriseAssignedAccess configuration service provider enables the IT administrator to provision the device into a state with locked down user experience. You can customize the start screen with a variety of

pinned applications, disable system buttons, configure buttons to have custom launch actions, and customize the settings panel to have only specific settings options available to the user.

WARNING

This feature should only be used on devices that are owned or provided by the enterprise company or organization or on a user owned device where the user allowed the device to be fully managed by the enterprise company.

As a Mobile Device Management Solutions Vendor, you must provide the following disclaimer to the IT administrator prior to the use of the feature.

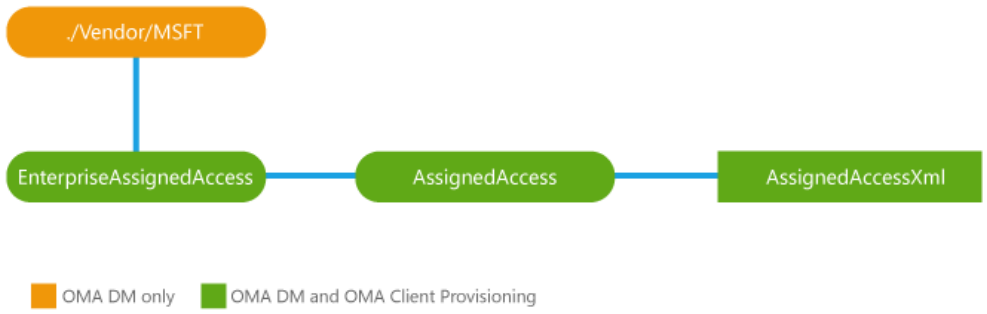
This feature may cause the device to fail or lose connectivity and require that the device be serviced at a Nokia-authorized repair center to reset to factory settings. Microsoft is not liable for any damage to the device or any loss of productivity that results from use of this feature. Microsoft requires that software vendors provide disclaimers to users when their products expose this feature and capabilities.

PLEASE READ ALL NOTES ASSSOCIATED WITH ASSIGN ACCESS BEFORE PROCEEDING:

- Assigned access should be used together with application allow/deny lists and restrictive policy controls. You may be able to deep link into settings or applications based on file handling. The combination of PolicyManager settings management, application management, and lockdown helps create the most secure locked down experience.
- Once an EnterpriseAssignedAccess has been provisioned to a device, the only way to remove this functionality is to reset the device to factory settings through Settings->about->reset or through hardware key combinations.
- It is possible that specific combinations of EnterpriseAssignedAccess and device policies through PolicyManager may render the device unusable. For example, if hardware key reset combinations are disabled, the device must be sent to a factory-authorized repair center for repair to remove EnterpriseAssignedAccess functionality.
- On every reboot, the AssignedAccess XML will be reapplied and any user settings or options will be overridden. This includes Start Screen size (if the user selected "more tiles" for 6-column start, and the EnterpriseAssignedAccess feature requires 4-column start), then on reboot, the user's start screen will be forced back to a 4-column start and their start screen arrangement will be converted down back to a 4-column start.
- Email accounts cannot be managed, and those will always be displayed on the All Apps page when EnterpriseAssignedAccess is configured. This can be addressed by blocking provisioning of email accounts using PolicyManager.
- Family Rooms cannot be blocked from view if they are previously pinned to the Start Screen. This can be addressed by blocking provisioning of a Microsoft Account connection using PolicyManager.
- It is only possible to block the Store through the tile. If a user searches their apps list, an item is shown that lets the user to "Search Store," which enables the user to deep link into the Store directly even though the application does not exist. Additionally, this allows users to purchase against any provisioned Microsoft Account. This can be addressed by blocking access to the Store and blocking provisioning of a Microsoft Account using PolicyManager.

- All settings can be accessed through deep links. This includes apps that deep link into settings pages and QR codes/NFC Tags that can deep link into apps or settings pages. Users can be blocked from changing deep-linked settings by disabling policies PolicyManager. You cannot allow Wi-Fi, but prevent the user from changing the state to prevent the user from turning off Wi-Fi.
- If action+notification center is usable, settings and apps may be directly deep linked from this experience. This can be addressed by blocking access to apps using PolicyManager and Allow/Deny lists.
- Any application that has a web browser control can deep link into the Microsoft Store and purchase apps against any provisioned Microsoft Account. If a Microsoft Account is not present, one can be added, even if the settings page is not visible. This can be addressed by blocking access to the Store and Microsoft Account provisioning using PolicyManager.
- Apps that have web links can launch Internet Explorer, enabling a full browser experience, even if Internet Explorer is not visible. This can be addressed by blocking access to Internet Explorer using PolicyManager.
- Any app that is visible in a user-action required state allows deep linking into the Store. This can be addressed by blocking access to the Store using PolicyManager.
- Internet Explorer cache and back stack are not cleared when provisioning EnterpriseAssignedAccess.

The following diagram shows the EnterpriseAssignedAccess configuration service provided object in a tree format.



AssignedAccess

The parent node for the AssignedAccess XML.

AssignedAccess XML

The XML code that controls the assigned access settings applied to the device. Supported operations are Add, Get, and Replace.

First party application Product IDs

In order to pin or allow certain first party applications, application Product IDs are required to configure their placement. Third party Store application Product IDs can be found on windowsphone.com

The list of <Apps> in the AssignedAccess XML is an allow list of applications. As a result, if no apps are included in the list, no apps will be visible on Start. The following table shows the applications and their product IDs.

Application	productId
Alarms	{5B04B775-356B-4AA0-AAF8-6491FFEA560A}
Battery Saver	{C551F76F-3368-42BB-92DF-7BFBB9265636}
Bing Finance	{1E0440F1-7ABF-4B9A-863D-177970EEFB5E}
Bing Food	{CC512389-0456-430F-876B-704B17317DE2}
Bing Health	{CBB8C3BD-99E8-4176-AD8C-95EC6A3641C2}
Bing News	{9C3E8CAD-6702-4842-8F61-B8B33CC9CAF1}
Bing Sports	{0F4C8C7E-7114-4E1E-A84C-50664DB13B17}
Bing Travel	{19CD0687-980B-4838-8880-5F68ABA1671E}
Bing Weather	{63C2A117-8604-44E7-8CEF-DF10BE3A57C8}
Calculator	{5B04B775-356B-4AA0-AAF8-6491FFEA5603}
Calendar	{36F9FA1C-FDAD-4CF0-99EC-C03771ED741A}
Camera (built-in)	{5B04B775-356B-4AA0-AAF8-6491FFEA5631}
Cortana	{5B04B775-356B-4AA0-AAF8-6491FFEA568C}
Data Sense	{5B04B775-356B-4AA0-AAF8-6491FFEA5646}
Email	{5B04B775-356B-4AA0-AAF8-6491FFEA5614}
Facebook	{0C340A67-3288-4C76-9375-0F2FEFBA0412}
Games	{50A6AEF0-4F35-434B-9308-CB3251303AE4}
Internet Explorer	{5B04B775-356B-4AA0-AAF8-6491FFEA5666}
Maps	{5B04B775-356B-4AA0-AAF8-6491FFEA5686}
Messaging	{5B04B775-356B-4AA0-AAF8-6491FFEA5610}
Music	{D2B6A184-DA39-4C9A-9E0A-8B589B03DEC0}
Office Hub	{5B04B775-356B-4AA0-AAF8-6491FFEA561E}
OneDrive	{AD543082-80EC-45BB-AA02-FFE7F4182BA8}
OneNote	{5B04B775-356B-4AA0-AAF8-6491FFEA561B}
People	{5B04B775-356B-4AA0-AAF8-6491FFEA5615}
Phone	{5B04B775-356B-4AA0-AAF8-6491FFEA5611}
Photos	{5B04B775-356B-4AA0-AAF8-6491FFEA5632}
Podcasts	{C3215724-B279-4206-8C3E-61D1A9D63ED3}
Settings	{5B04B775-356B-4AA0-AAF8-6491FFEA5601}
Storage Sense	{5B04B775-356B-4AA0-AAF8-6491FFEA564D}
Store	{5B04B775-356B-4AA0-AAF8-6491FFEA5633}
Video	{6AFFE59E-0467-4701-851F-7AC026E21665}
Wallet	{5B04B775-356B-4AA0-AAF8-6491FFEA5683}

See the following sample.

Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <Apps>
      <!-- Alarms -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA560A}">
```

```
        <PinToStart>
          <Size>Medium</Size>
          <Location>
            <LocationX>0</LocationX>
            <LocationY>4</LocationY>
          </Location>
        </PinToStart>
      </Application>
    </Apps>
  </Default>
</HandheldLockdown>
```

Button lockdown +remap

Buttons can be locked down to prevent the button from being executing or starting their normal functionality. Additionally, button functionality can be remapped to do specific functionality like launching an application.

Button (on device)	Button XML name	Press | PressAndHold Block/Override	Can be remapped
Camera	Camera	Block and Override	No
Back	Back	Not supported	No
Start (Windows Key)	Start	Block and Override	No
Search	Search	Block and Override	Yes (App launch)
Volume Up	---	Not supported	No
Volume Down	---	Not supported	No
Power	---	Not supported	No

In order to lock down all button presses, all buttons must be added to <ButtonLockDownList> with both ButtonEvent types added – Press and PressAndHold. See the following sample.

Button Sample XML Excerpt: Lockdown and remapping

Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <Buttons>
      <ButtonLockdownList>
        <!-- Lockdown all buttons -->
        <Button name="Search">
        </Button>
        <Button name="Camera">
          <ButtonEvent name="Press" />
          <ButtonEvent name="PressAndHold" />
        </Button>
      </ButtonLockdownList>
      <ButtonRemapList>
        <Button name="Search">
          <ButtonEvent name="Press">
            <!-- Settings -->
            <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}" parameters="" />
          </ButtonEvent>
        </Button>
      </ButtonRemapList>
    </Buttons>
```

```
  </Default>
</HandheldLockdown>
```

Button Sample XML Excerpt: lockdown

Here is a sample of the AssignedAccess XML that locks down the default Camera button (both press and press and hold) remaps the Search press button to launch the Settings application.

Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <Buttons>
      <ButtonLockdownList>
        <Button name="Camera">
          <ButtonEvent name="Press" />
          <ButtonEvent name="PressAndHold" />
        </Button>
        <ButtonRemapList>
          <Button name="Search">
            <ButtonEvent name="Press">
              <!-- Settings -->
              <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}" parameters="" />
            </ButtonEvent>
          </Button>
        </ButtonRemapList>
      </ButtonLockdownList>
    </Buttons>
  </Default>
</HandheldLockdown>
```

Settings: System + Application settings lockdown

Several settings + application items can be managed to configure whether those options can be viewed in the settings page. Note that OEMs can configure additional settings CPLs, and these are not included in this list.

NOTE: the list of settings/applications are an allow list. This means, if no settings are included in the AssignedAccess XML, then no settings/applications will be listed.

Settings page	Settings name	name
System	about	Microsoft.About
System	ease of access	Microsoft.Accessibility
System	email+accounts	Microsoft.Accounts
System	advertising id	Microsoft.AdvertisingId
System	airplane mode	Microsoft.AirplaneMode
System	battery saver	Microsoft.BatterySaver
System	Bluetooth	Microsoft.Bluetooth
System	brightness	Microsoft.Brightness
System	Cellular+SIM	Microsoft.CellularConn
System	backup	Microsoft.CloudStorageCpl
System	workplace	Microsoft.CompanyAccount
System	date+time	Microsoft.DateTime
System	quiet hours	Microsoft.DoNotDisturb
System	driving mode	Microsoft.DrivingMode

System	feedback	Microsoft.Feedback
System	find my phone	Microsoft.FindMyPhone
System	kids corner	Microsoft.KidZone
System	language	Microsoft.Language
System	location	Microsoft.Location
System	project my screen	Microsoft.MirrorUX
System	notifications+actions	Microsoft.NocenterSettings
System	lock screen	Microsoft.PhoneLock
System	region	Microsoft.Regional
System	sync my settings	Microsoft.RoamingCpl
System	screen rotation	Microsoft.RotationLock
System	internet sharing	Microsoft.SoftAP
System	ringtones+sounds	Microsoft.Sounds
System	speech	Microsoft.Speech
System	storage sense	Microsoft.StorageSettings
System	start+theme	Microsoft.Themes
System	keyboard	Microsoft.TouchKeyboard
System	phone update	Microsoft.Updates
System	VPN	Microsoft.VPN
System	Wi-Fi	Microsoft.WiFi
System	NFC	Microsoft.NFC
System	USB	Microsoft.USB
System	data sense	Microsoft.DataSmart
Application	internet explorer	Microsoft.IE
Application	maps	Microsoft.Maps
Application	messaging	Microsoft.Messaging
Application	Office	Microsoft.OfficeMobile
Application	people	Microsoft.Contacts
Application	phone	Microsoft.Phone
Application	Photos+camera	Microsoft.Photos
Application	search	Microsoft.Search
Application	store	Microsoft.Marketplace
Application	wallet	Microsoft.Wallet
Application	cortana	Microsoft.AssistUX

Settings
Required field.

Settings\System
Optional field. Used as an allow list of allowed settings that are displayed under System

Settings\Application
Optional. An allow list of allowed settings that show up under Application. See the previous table .

Settings Sample XML Excerpt

The following sample AssignedAccess XML locks down the settings page to only show two settings items in the Settings application: system\about and application\phone. Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <Settings>
      <System name="Microsoft.About" />
      <Application name="Microsoft.Phone" />
    </Settings>
  </Default>
</HandheldLockdown>
```

Action Center

Action Center includes both quick settings and notifications that users can quickly access. You can manage this feature.

ActionCenter
Required. Supports enabled="true" or enabled="false".

Action Center Sample XML Excerpt

Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <ActionCenter enabled="true" />
  </Default>
</HandheldLockdown>
```

Menu items

The start screen allows for a menu to be shown to help the user configure and customize their start screen. The menu can be triggered by pressing and holding start screen applications or tiles. This includes resizing tiles, moving their placement, and pinning additional tiles to start. To prevent this experience from being exposed in order to more fully lockdown this experience, Menu Items can be disabled.

MenuItems
Required.

MenuItems\DisableMenuItems
Optional. Disables all menu items on modern shell including long press on applications to prevent menu items from being displayed.

Disable Menu Items Sample XML Excerpt

Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <MenuItems>
      <DisableMenuItems/>
    </MenuItems>
  </Default>
</HandheldLockdown>
```

Start Screen size

Three Start screen configurations are supported: small, medium, large. Medium and large represent 3-column start views, which enable six small tiles to be pinned in one row. The main difference between medium and large start screen scaling is resolution and keyboard scaling. Medium is recommended for all 720P and lower screen resolutions. Large is recommended for 1080P screen resolutions.

StartScreenSize
Required. Supports values of "Small", "Medium", or "Large".

Start Screen Size Sample XML Excerpt

Note that all top-level fields under <Default> must be included as part of the XML, unlike the following sample excerpt which does not show include other top-level fields.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <StartScreenSize>Small</StartScreenSize>
  </Default>
</HandheldLockdown>
```

Sample AssignedAccess XML

The following is a sample AssignedAccess XML. Note that this must be escaped prior to being provisioned through SyncML.

NOTE: PolicyManager policy settings should be used together with lockdown XML to provide blocking access through deep linking. Lockdown only blocks user-facing pieces of the experience, and does not prevent users from deep linking into applications, both first and thrid party, and settings pages.

NOTE: All top-level fields under <Default> are required to be included, even if no sub-fields are used.

NOTE: Please ensure that the XML is wrapped inside a SyncML body payload when it is provisioned to the device

NOTE: The XML should be placed in the <Data> field and either fully escaped or use <Data>![CDATA[<*insert_xml_here*>]]</Data> for wrapping unescaped data.

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
  <Default>
    <ActionCenter enabled="true" />
    <Apps>
      <!-- Alarms -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA560A}" />
      <!-- Battery Saver -->
      <Application productId="{C551F76F-3368-42BB-92DF-7BFBB9265636}" />
      <!-- Bing Finance -->
      <Application productId="{1E0440F1-7ABF-4B9A-863D-177970EEFB5E}" />
      <!-- Bing Food -->
      <Application productId="{CC512389-0456-430F-876B-704B17317DE2}" />
      <!-- Bing Health -->
      <Application productId="{CBB8C3BD-99E8-4176-AD8C-95EC6A3641C2}" />
      <!-- Bing News -->
      <Application productId="{9C3E8CAD-6702-4842-8F61-B8B33CC9CAF1}" />
      <!-- Bing Sports -->
      <Application productId="{0F4C8C7E-7114-4E1E-A84C-50664DB13B17}" />
      <!-- Bing Travel -->
      <Application productId="{19CD0687-980B-4838-8880-5F68ABA1671E}" />
      <!-- Bing Weather -->
```

```
      <Application productId="{63C2A117-8604-44E7-8CEF-DF10BE3A57C8}" />
      <!-- Calculator -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5603}" />
      <!-- Calendar -->
      <Application productId="{36F9FA1C-FDAD-4CF0-99EC-C03771ED741A}">
        <PinToStart>
          <Size>Medium</Size>
          <Location>
            <LocationX>0</LocationX>
            <LocationY>4</LocationY>
          </Location>
        </PinToStart>
      </Application>
      <!-- Camera -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5631}" />
      <!-- Cortana -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA568C}">
        <PinToStart>
          <Size>Medium</Size>
          <Location>
            <LocationX>0</LocationX>
            <LocationY>2</LocationY>
          </Location>
        </PinToStart>
      </Application>
      <!-- Data Sense -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5646}" />
      <!-- Email -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5614}">
        <PinToStart>
          <Size>Small</Size>
          <Location>
            <LocationX>0</LocationX>
            <LocationY>1</LocationY>
          </Location>
        </PinToStart>
      </Application>
      <!-- Facebook -->
      <Application productId="{0C340A67-3288-4C76-9375-0F2FEFBA0412}" />
      <!-- Games -->
      <Application productId="{50A6AEF0-4F35-434B-9308-CB3251303AE4}" />
      <!-- Internet Explorer -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5660}">
        <PinToStart>
          <Size>Small</Size>
          <Location>
            <LocationX>1</LocationX>
            <LocationY>1</LocationY>
          </Location>
        </PinToStart>
      </Application>
      <!-- Maps -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5686}" />
      <!-- Messaging -->
      <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5610}">
        <PinToStart>
          <Size>Small</Size>
          <Location>
            <LocationX>1</LocationX>
            <LocationY>0</LocationY>
          </Location>
        </PinToStart>
```

```
    </Application>
    <!-- Music -->
    <Application productId="{D2B6A184-DA39-4C9A-9E0A-8B589B03DEC0}">
      <PinToStart>
        <Size>Medium</Size>
        <Location>
          <LocationX>2</LocationX>
          <LocationY>4</LocationY>
        </Location>
      </PinToStart>
    </Application>
    <!-- Office Hub -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA561E}" />
    <!-- OneDrive -->
    <Application productId="{AD543082-80EC-45BB-AA02-FFE7F4182BA8}" />
    <!-- OneNote -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA561B}" />
    <!-- People -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5615}">
      <PinToStart>
        <Size>Medium</Size>
        <Location>
          <LocationX>2</LocationX>
          <LocationY>0</LocationY>
        </Location>
      </PinToStart>
    </Application>
    <!-- Phone -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5611}">
      <PinToStart>
        <Size>Small</Size>
        <Location>
          <LocationX>0</LocationX>
          <LocationY>0</LocationY>
        </Location>
      </PinToStart>
    </Application>
    <!-- Photos -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5632}">
      <PinToStart>
        <Size>Large</Size>
        <Location>
          <LocationX>0</LocationX>
          <LocationY>2</LocationY>
        </Location>
      </PinToStart>
    </Application>
    <!-- Podcast -->
    <Application productId="{C3215724-B279-4206-8C3E-61D1A9D63ED3}" />
    <!-- Settings -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}" />
    <!-- Storage Sense -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA564D}" />
    <!-- Store -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5633}">
      <PinToStart>
        <Size>Medium</Size>
        <Location>
          <LocationX>2</LocationX>
          <LocationY>2</LocationY>
        </Location>
      </PinToStart>
```

```
    </Application>
    <!-- Video -->
    <Application productId="{6AFFE59E-0467-4701-851F-7AC026E21665}" />
    <!-- Wallet -->
    <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5683}"/>
  </Apps>
  <Buttons>
    <ButtonLockdownList>
      <!-- Lockdown all buttons -->
      <Button name="Search">
      </Button>
      <Button name="Camera">
        <ButtonEvent name="Press" />
        <ButtonEvent name="PressAndHold" />
      </Button>
    </ButtonLockdownList>
    <ButtonRemapList>
      <Button name="Search">
        <ButtonEvent name="Press">
          <!-- Settings -->
          <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}" parameters="" />
        </ButtonEvent>
      </Button>
    </ButtonRemapList>
  </Buttons>
  <MenuItems>
    <DisableMenuItems/>
  </MenuItems>
  <Settings>
    <System name="Microsoft.About" />
    <System name="Microsoft.Accessibility" />
    <System name="Microsoft.Accounts" />
    <System name="Microsoft.AdvertisingId" />
    <System name="Microsoft.AirplaneMode" />
    <System name="Microsoft.AssistUX" />
    <System name="Microsoft.BatterySaver" />
    <System name="Microsoft.Bluetooth" />
    <System name="Microsoft.Brightness" />
    <System name="Microsoft.CellularConn" />
    <System name="Microsoft.CloudStorageCpl" />
    <System name="Microsoft.CompanyAccount" />
    <System name="Microsoft.DateTime" />
    <System name="Microsoft.DoNotDisturb" />
    <System name="Microsoft.DrivingMode" />
    <System name="Microsoft.Feedback" />
    <System name="Microsoft.FindMyPhone" />
    <System name="Microsoft.KidZone" />
    <System name="Microsoft.Language" />
    <System name="Microsoft.Location" />
    <System name="Microsoft.MirrorUX" />
    <System name="Microsoft.NocenterSettings" />
    <System name="Microsoft.PhoneLock" />
    <System name="Microsoft.ProfileUpdate" />
    <System name="Microsoft.Proximity" />
    <System name="Microsoft.Regional" />
    <System name="Microsoft.RoamingCpl" />
    <System name="Microsoft.RotationLock" />
    <System name="Microsoft.SoftAP" />
    <System name="Microsoft.Sounds" />
    <System name="Microsoft.Speech" />
    <System name="Microsoft.StorageSettings" />
    <System name="Microsoft.Themes" />
```

```
      <System name="Microsoft.TouchKeyboard" />
      <System name="Microsoft.Updates" />
      <System name="Microsoft.VPN" />
      <System name="Microsoft.WiFi" />
      <Application name="Microsoft.Search" />
      <Application name="Microsoft.IE" />
      <Application name="Microsoft.Maps" />
      <Application name="Microsoft.Messaging" />
      <Application name="Microsoft.OfficeMobile" />
      <Application name="Microsoft.Contacts" />
      <Application name="Microsoft.Phone" />
      <Application name="Microsoft.Photos" />
      <Application name="Microsoft.Search" />
      <Application name="Microsoft.Marketplace" />
      <Application name="Microsoft.Wallet" />
      <Application name="Microsoft.AssistUX" />
    </Settings>
    <StartScreenSize>Small</StartScreenSize>
  </Default>
</HandheldLockdown>
```

Sample AssignedAccess SyncML

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Add>
      <CmdID>1</CmdID>
      <Item>
        <Target>

<LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/AssignedAccess/AssignedAccessXml</LocURI>
        </Target>
        <Meta>
          <Format xmlns="syncml:metinf">chr</Format>
        </Meta>
        <Data>![CDATA[insert_xml_here]]</Data>
      </Item>
    </Add>
    <Final/>
  </SyncBody>
</SyncML>
```

Schema for AssignedAccess XML

This XSD can be used to validate that the XML in the <Data> block is a valid XML that can be provisioned successfully onto the device

NOTE: The following features are **unsupported** for Windows Phone 8.1

- Role Lists
- CSP Runner
- Button management for: Custom1, Custom2, Custom3

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
>
  <!-- COMPLEX TYPE: ROLE LIST TYPE -->
  <xs:complexType name="role_list_t">
    <xs:sequence minOccurs="1" maxOccurs="1">
      <xs:element name="Role" type="role_t" minOccurs="1" maxOccurs="unbounded"/>
```

```
    </xs:sequence>
  </xs:complexType>
  <!-- COMPLEX TYPE: START SCREEN SIZE TYPE -->
  <xs:simpleType name="startscreen_size_t">
    <xs:restriction base="xs:string">
      <!-- Small: 4 columns-->
      <xs:enumeration value="Small"/>
      <!-- Large: 6 columns-->
      <xs:enumeration value="Large"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- COMPLEX TYPE: APPLICATION LIST TYPE -->
  <xs:complexType name="application_list_t">
    <xs:sequence minOccurs="0" maxOccurs="1">
      <xs:element name="Application" type="application_t" minOccurs="0" maxOccurs="unbounded"
/>
    </xs:sequence>
  </xs:complexType>
  <!-- COMPLEX TYPE: BUTTON LIST TYPE -->
  <xs:complexType name="button_list_t">
    <xs:sequence minOccurs="0" maxOccurs="1">
      <xs:element name="Button" minOccurs="0" maxOccurs="6" type="button_t"/>
    </xs:sequence>
  </xs:complexType>
  <!-- COMPLEX TYPE: MENU ITEM LIST TYPE -->
  <xs:complexType name="menu_item_list_t">
    <xs:sequence minOccurs="0" maxOccurs="1">
      <xs:element name="DisableMenuItems" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!-- COMPLEX TYPE: DEFAULT TYPE -->
  <xs:complexType name="default_basic_t">
    <xs:sequence minOccurs="1">
      <xs:element name="ActionCenter" type="actioncenter_t" minOccurs="1"/>
      <xs:element name="Apps" type="application_list_t" minOccurs="1">
        <xs:unique name="duplicateAppsForbidden">
          <xs:selector xpath="Application"/>
          <xs:field xpath="@productId"/>
        </xs:unique>
      </xs:element>
      <xs:element name="Buttons" minOccurs="1">
        <xs:complexType>
          <xs:all>
            <xs:element name="ButtonLockdownList" type="button_list_t" minOccurs="0"/>
            <xs:element name="ButtonRemapList" type="button_list_t" minOccurs="0"/>
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="CSPRunner" minOccurs="0"/>
      <xs:element name="MenuItems" type="menu_item_list_t" minOccurs="1"/>
      <xs:element name="Settings" minOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="System" type="setting_t" minOccurs="0" maxOccurs="unbounded" />
            <xs:element name="Application" type="setting_t" minOccurs="0"
maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

```
<!-- COMPLEX TYPE: ROLE TYPE -->
<xs:complexType name="role_t">
  <xs:complexContent>
    <xs:extension base="default_basic_t">
      <xs:attribute name="guid" type="guid_t" use="required"/>
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- COMPLEX TYPE: DEFAULT ROLE TYPE -->
<xs:complexType name="default_role_t">
  <xs:complexContent>
    <xs:extension base="default_basic_t">
      <xs:sequence minOccurs="1">
        <xs:element name="StartScreenSize" type="startscreen_size_t" minOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- COMPLEX TYPE: Action Center -->
<xs:complexType name="actioncenter_t">
  <xs:attribute type="xs:boolean" name="enabled" use="required"/>
</xs:complexType>
<!-- COMPLEX TYPE: APPLICATION TYPE -->
<xs:complexType name="application_t">
  <xs:all minOccurs="0">
    <xs:element name="PinToStart" type="start_tile_t" />
  </xs:all>
  <xs:attribute name="productId" type="guid_t" use="required"/>
  <xs:attribute name="parameters" type="xs:string" use="optional"/>
  <xs:attribute name="autoRun" type="xs:boolean" use="optional"/>
</xs:complexType>
<!-- COMPLEX TYPE: START SCREEN TILE CONFIGURATION TYPE-->
<xs:complexType name="start_tile_t">
  <xs:all minOccurs="1" maxOccurs="1">
    <xs:element name="Size" type="tile_size_t" minOccurs="1" />
    <xs:element name="Location" type="tile_location_t" minOccurs="1" />
  </xs:all>
</xs:complexType>
<!-- COMPLEX TYPE: SETTING TYPE -->
<xs:complexType name="setting_t">
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
<!-- COMPLEX TYPE: BUTTON TYPE -->
<xs:complexType name="button_t">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="ButtonEvent" type="button_event_t" minOccurs="0" maxOccurs="2"/>
  </xs:sequence>
  <xs:attribute name="name" type="supported_button_t" use="required"/>
</xs:complexType>
<!-- COMPLEX TYPE: BUTTON EVENT TYPE -->
<xs:complexType name="button_event_t">
  <xs:all minOccurs="0" maxOccurs="1">
    <xs:element name="Application" type="application_t" minOccurs="0" maxOccurs="1" />
  </xs:all>
  <xs:attribute name="name" type="supported_button_event_t" use="required"/>
</xs:complexType>

<!--COMPLEX TYPE: START TILE TYPE-->
<xs:complexType name="tile_location_t">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="LocationX" type="xs:unsignedLong"/>
```

```
      <xs:element name="LocationY" type="xs:unsignedLong"/>
    </xs:sequence>
  </xs:complexType>

  <!-- SIMPLE TYPE: SUPPORTED BUTTON TYPE -->
  <xs:simpleType name="supported_button_t">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Start"/>
      <xs:enumeration value="Search"/>
      <xs:enumeration value="Camera"/>
      <xs:enumeration value="Custom1"/>
      <xs:enumeration value="Custom2"/>
      <xs:enumeration value="Custom3"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- SIMPLE TYPE: SUPPORTED BUTTON EVENT TYPE -->
  <xs:simpleType name="supported_button_event_t">
    <xs:restriction base="xs:string">
      <xs:enumeration value="All"/>
      <xs:enumeration value="Press"/>
      <xs:enumeration value="PressAndHold"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- SIMPLE TYPE: GUID -->
  <xs:simpleType name="guid_t">
    <xs:restriction base="xs:string">
      <xs:pattern value="\{[0-9a-fA-F]{8}\-([0-9a-fA-F]{4}\-){3}[0-9a-fA-F]{12}\}"/>
    </xs:restriction>
  </xs:simpleType>
  <!--SIMPLE TYPE: TILE SIZE-->
  <xs:simpleType name="tile_size_t">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Small"/>
      <xs:enumeration value="Medium"/>
      <xs:enumeration value="Large"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- SCHEMA -->
  <xs:element name="HandheldLockdown">
    <xs:complexType>
      <xs:all minOccurs="1">
        <xs:element name="Default" type="default_role_t"/>
        <xs:element name="RoleList" type="role_list_t" minOccurs="0">
          <xs:unique name="duplicateRolesForbidden">
            <xs:selector xpath="Role"/>
            <xs:field xpath="@guid"/>
          </xs:unique>
        </xs:element>
      </xs:all>
      <xs:attribute name="version" use="required" type="xs:decimal"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Windows Embedded 8.1 Handheld device management

Windows Embedded 8.1 Handheld uses the same provisioning and device management model as Windows Phone, so your knowledge of Windows Phone transfers directly to Handheld 8.1. In addition, Handheld 8.1 introduces assigned access, which is a suite of features that allows an enterprise to lock down the user experience of the device platform.

For more information about assigned access, see the Administrator Guide for Windows Embedded 8.1 Handheld.

The provisioning XML file (Handheld 8.1)

The provisioning XML file (Prov.xml) for Windows Embedded 8.1 Handheld contains the configuration settings and lockdown information for the enterprise devices. It can be pushed to a device by using the mobile device management (MDM) service and then restarting the device. It can also be sideloaded by using a near field communication (NFC) tag, SD card, or other data source (such as a bar code scanner), and then applying during the out-of-box experience (OOBE).

The Prov.XML contains the following:

- (Required) The requested certificates.
- (Required) DM client configuration.
- (Required) Wireless must be provisioned.
- (Optional) An enterprise application token and an enterprise app download link to allow the enrollment client to download a Company Hub app or enterprise app at the end of enrollment.
- (Optional) Assigned access XML.

The following table shows how you can use configuration service providers to configure devices.

Configuration service provider	Description
EnterpriseExt configuration service provider	Allows the enterprise to use the MDM service to enroll devices to the MDM server in an enterprise environment, restart a device, and manage the maintenance window schedule for devices so that they can perform device updates and other management tasks.
EnterpriseAssignedAccess configuration service provider	Allows the enterprise to use Windows Embedded 8.1 Handheld features to configure custom layouts on a device. For example, the administrator can lock down a device so that only apps specified in an Allow list are available. Apps not on the Allow list remain installed on the device, but are hidden from view.
EnterpriseExtFileSystem configuration service provider	Allows IT administrators to add, retrieve, or change files in the file system through the MDM service. For example, you can use this configuration service provider to push a provisioning

To create a Prov.xml file to configure devices

1. Using a text or XML editor, copy the sample OMA Client Provisioning XML from this topic and save it to a new XML file.
2. Using the examples in the Configuration service provider reference topics, change the values to the appropriate values for your organization.
3. Make sure that the provisioning file is encoded as UTF-8 or UTF-16LE, including the byte order mark (BOM).
4. (optional) Using MDM, encrypt Prov.xml. The encryption password will need to be provided during OOBE configuration. The encryption password can use only the following characters:
   - Uppercase letters of European languages (A through Z, with diacritic marks, Greek and Cyrillic characters)
   - Lowercase letters of European languages (a through z, sharp-s, with diacritic marks, Greek and Cyrillic characters)
   - Base 10 digits (0 through 9)
   - Nonalphanumeric characters: ~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/
5. Do one of the following:
   - If you will push the Prov.xml to devices by using the MDM service, save the file to your development computer, and then push it to a device.
   - If you will use an SD card or other data source (such as a bar code scanner), save the Prov.xml file to the root directory of the data source so that the IT administrator can sideload it, and then apply it during OOBE.
   - If you will use an NFC tag or device, make sure that you followed the components of an NFC tag as described in Enable near field communication. Save the Prov.xml file to the NFC tag or device so that the IT administrator can sideload it, and then apply it during OOBE.

Sample OMA Client Provisioning

The following example shows a provisioning XML file (Prov.xml) that is applied to a device during OOBE and that contains settings for a Wi-Fi connection, MDM enrollment, and profile lockdown.

Note: Formatting in this sample file uses escaped characters, such as < in place of <, as a result of XML embedded in XML. Do not replace the escaped characters.

```
<wap-provisioningdoc>
    <characteristic type="WiFi">
        <characteristic type="Profile">
            <characteristic type="Open">
                <parm name="WlanXml" datatype="string" value="&lt;?xml
version=&quot;1.0&quot;?&gt;&lt;WLANProfile
xmlns=&quot;http://www.microsoft.com/networking/WLAN/profile/v1&quot;&gt;&lt;name&gt;WIFI_OPEN
&lt;/name&gt;&lt;SSIDConfig&gt;&lt;SSID&gt;&lt;name&gt;WIFI_OPEN&lt;/name&gt;&lt;/SSID&gt;&lt;
/SSIDConfig&gt;&lt;connectionType&gt;ESS&lt;/connectionType&gt;&lt;connectionMode&gt;auto&lt;/
connectionMode&gt;&lt;MSM&gt;&lt;security&gt;&lt;authEncryption&gt;&lt;authentication&gt;open&
lt;/authentication&gt;&lt;encryption&gt;none&lt;/encryption&gt;&lt;/authEncryption&gt;&lt;/sec
urity&gt;&lt;/MSM&gt;&lt;/WLANProfile&gt;" />
            </characteristic>
        </characteristic>
    </characteristic>
    <characteristic type="EnterpriseExt">
        <characteristic type="MDM">
            <parm name="Server" value="https://localhost:443" />
```

```
            <parm name="Username" value="username@contoso.com" />
            <parm name="Password" value="password" />
            <parm name="EnableDeviceEnrollment" value="false" datatype="boolean" />
        </characteristic>
    </characteristic>
    <characteristic type="EnterpriseAssignedAccess">
        <characteristic type="AssignedAccess">
            <parm name="AssignedAccessXml" datatype="string" value="&lt;?xml
version=&quot;1.0&quot; encoding=&quot;utf-8&quot;?&gt;
                &lt;HandheldLockdown version=&quot;1.0&quot;&gt;
                    &lt;Default&gt;
                        &lt;ActionCenter enabled=&quot;true&quot; /&gt;
                            &lt;Apps&gt;
                                &lt;!-- Phone App --&gt;
                                    &lt;Application productId=&quot;{5B04B775-356B-4AA0-AAF8-
6491FFEA5611}&quot;&gt;
                                        &lt;PinToStart&gt;
                                            &lt;Size&gt;Medium&lt;/Size&gt;
                                            &lt;Location&gt;
                                                &lt;LocationX&gt;0&lt;/LocationX&gt;
                                                &lt;LocationY&gt;0&lt;/LocationY&gt;
                                            &lt;/Location&gt;
                                        &lt;/PinToStart&gt;
                                    &lt;/Application&gt;
                                    &lt;!-- Settings --&gt;
                                    &lt;Application productId=&quot;{5B04B775-356B-4AA0-AAF8-
6491FFEA5601}&quot;&gt;
                                        &lt;PinToStart&gt;
                                            &lt;Size&gt;Medium&lt;/Size&gt;
                                            &lt;Location&gt;
                                                &lt;LocationX&gt;2&lt;/LocationX&gt;
                                                &lt;LocationY&gt;0&lt;/LocationY&gt;
                                            &lt;/Location&gt;
                                        &lt;/PinToStart&gt;
                                    &lt;/Application&gt;
                            &lt;/Apps&gt;
                            &lt;Buttons&gt;
                                &lt;ButtonLockdownList&gt;
                                    &lt;!-- Lockdown all buttons except Search --&gt;
                                    &lt;Button name=&quot;Search&quot;&gt;
                                    &lt;/Button&gt;
                                    &lt;Button name=&quot;Camera&quot;&gt;
                                        &lt;ButtonEvent name=&quot;Press&quot; /&gt;
                                        &lt;ButtonEvent name=&quot;PressAndHold&quot; /&gt;
                                    &lt;/Button&gt;
                                    &lt;Button name=&quot;Custom1&quot;&gt;
                                        &lt;ButtonEvent name=&quot;Press&quot; /&gt;
                                        &lt;ButtonEvent name=&quot;PressAndHold&quot; /&gt;
                                    &lt;/Button&gt;
                                    &lt;Button name=&quot;Custom2&quot;&gt;
                                        &lt;ButtonEvent name=&quot;Press&quot; /&gt;
                                        &lt;ButtonEvent name=&quot;PressAndHold&quot; /&gt;
                                    &lt;/Button&gt;
                                    &lt;Button name=&quot;Custom3&quot;&gt;
                                        &lt;ButtonEvent name=&quot;Press&quot; /&gt;
                                        &lt;ButtonEvent name=&quot;PressAndHold&quot; /&gt;
                                    &lt;/Button&gt;
                                &lt;/ButtonLockdownList&gt;
                                &lt;ButtonRemapList&gt;
                                    &lt;Button name=&quot;Search&quot;&gt;
                                        &lt;ButtonEvent name=&quot;Press&quot;&gt;
                                            &lt;!-- TicTapToe --&gt;
```

```
                                        &lt;Application productId=&quot;{08179793-ED2E-
45EA-BA12-BDE3EE9C3CE3}&quot; parameters=&quot;&quot; /&gt;
                                    &lt;/ButtonEvent&gt;
                                &lt;/Button&gt;
                            &lt;/ButtonRemapList&gt;
                        &lt;/Buttons&gt;
                        &lt;MenuItems&gt;
                            &lt;DisableMenuItems /&gt;
                        &lt;/MenuItems&gt;
                        &lt;Settings&gt;
                            &lt;System name=&quot;Microsoft.About&quot; /&gt;
                            &lt;System name=&quot;Microsoft.NocenterSettings&quot; /&gt;
                            &lt;System name=&quot;Microsoft.CompanyAccount&quot; /&gt;
                        &lt;/Settings&gt;
                        &lt;StartScreenSize&gt;Small&lt;/StartScreenSize&gt;
                    &lt;/Default&gt;
                    &lt;RoleList&gt;
                        &lt;Role guid=&quot;{88501844-3B51-4C9F-9DA7-7CA745E7DA6B}&quot;
name=&quot;Associate&quot;&gt;
                        &lt;ActionCenter enabled=&quot;false&quot; /&gt;
                            &lt;Apps&gt;
                                &lt;!-- Settings --&gt;
                                &lt;Application productId=&quot;{5B04B775-356B-4AA0-AAF8-
6491FFEA5601}&quot;&gt;
                                    &lt;PinToStart&gt;
                                        &lt;Size&gt;Medium&lt;/Size&gt;
                                        &lt;Location&gt;
                                            &lt;LocationX&gt;0&lt;/LocationX&gt;
                                            &lt;LocationY&gt;0&lt;/LocationY&gt;
                                        &lt;/Location&gt;
                                    &lt;/PinToStart&gt;
                                &lt;/Application&gt;
                            &lt;/Apps&gt;
                            &lt;Buttons&gt;
                                &lt;ButtonLockdownList&gt;
                                &lt;/ButtonLockdownList&gt;
                            &lt;/Buttons&gt;
                            &lt;MenuItems&gt;
                                &lt;DisableMenuItems /&gt;
                            &lt;/MenuItems&gt;
                            &lt;Settings&gt;
                                &lt;System name=&quot;Microsoft.About&quot; /&gt;
                                &lt;System name=&quot;Microsoft.Accessibility&quot; /&gt;
                                &lt;System name=&quot;Microsoft.Accounts&quot; /&gt;
                                &lt;System name=&quot;Microsoft.AdvertisingId&quot; /&gt;
                                &lt;System name=&quot;Microsoft.AirplaneMode&quot; /&gt;

                                &lt;System name=&quot;Microsoft.BatterySaver&quot; /&gt;
                                &lt;System name=&quot;Microsoft.Bluetooth&quot; /&gt;
                                &lt;System name=&quot;Microsoft.Brightness&quot; /&gt;
                                &lt;System name=&quot;Microsoft.CellularConn&quot; /&gt;
                                &lt;System name=&quot;Microsoft.CloudStorageCpl&quot;
/&gt;
                                &lt;System name=&quot;Microsoft.CompanyAccount&quot; /&gt;
                                &lt;System name=&quot;Microsoft.DateTime&quot; /&gt;
                                &lt;System name=&quot;Microsoft.DoNotDisturb&quot; /&gt;
                                &lt;System name=&quot;Microsoft.DrivingMode&quot; /&gt;
                                &lt;System name=&quot;Microsoft.Feedback&quot; /&gt;
                                &lt;System name=&quot;Microsoft.FindMyPhone&quot; /&gt;
                                &lt;System name=&quot;Microsoft.FlashAppSetting&quot;
/&gt;
                                &lt;System name=&quot;Microsoft.KidZone&quot; /&gt;
```

```
                                    &lt;System name=&quot;Microsoft.Language&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.Location&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.MirrorUX&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.NocenterSettings&quot;
/&gt;
                                    &lt;System name=&quot;Microsoft.PhoneLock&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.ProfileUpdate&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.Proximity&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.Regional&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.RoamingCpl&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.RotationLock&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.SoftAP&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.Sounds&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.Speech&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.StorageSettings&quot;
/&gt;
                                    &lt;System name=&quot;Microsoft.Themes&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.TouchKeyboard&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.Updates&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.USB&quot; /&gt;
                                    &lt;System name=&quot;Microsoft.VPN&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.AssistUX&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.Contacts&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.IE&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.Maps&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.Marketplace&quot;
/&gt;
                                    &lt;Application name=&quot;Microsoft.Messaging&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.OfficeMobile&quot;
/&gt;
                                    &lt;Application name=&quot;Microsoft.Phone&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.Photos&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.Search&quot; /&gt;
                                    &lt;Application name=&quot;Microsoft.Wallet&quot; /&gt;
                                &lt;/Settings&gt;
                            &lt;/Role&gt;
                        &lt;/RoleList&gt;
                    &lt;/HandheldLockdown&gt;" />
    </characteristic>
  </characteristic>
</wap-provisioningdoc>
```

Troubleshooting

A setting didn't provision properly during OOBE. Make sure that you followed the format guidelines for each characteristic node in the Prov.xml file. An incorrect format could cause a step to be skipped during the provisioning process with no error occurring. For information about correct formatting, see the specific configuration service provider topic.

Cryptography for prov.xml

The enterprise IT administrator will provide the encryption password when encrypting the provisioning file on the server side, and the device user will enter the same password when prompted. The password can be of any length (no minimum length is enforced on the device side), and is entered in a standard Windows Phone Splash password box. The password that is entered is used to generate a hash using the standard Crypto APIs. The internal algorithms are not exposed to the user and cannot be changed by the user.

Cryptographic algorithms and key lengths

Crypto APIs are used to encrypt and decrypt the provisioning file. After the key is taken as input from the user (Length 1 – 255), the hash key (CryptCreateHash) is generated using the SHA 256 algorithm, and the key (CryptDeriveKey) is derived using the AES 128 algorithm.

Algorithm	Key lengths
CALG_AES_128	Range [1, 255]
CALG_SHA_256	Range [1, 255]

Key management

The key is derived using the existing Crypto APIs. The CSP is MS_ENH_RSA_AES_PROV, and no container is needed (dwFlags is CRYPT_VERIFYCONTEXT). The key is derived using CryptDeriveKey where the algorithm is CALG_AES_128, and the hash is generated using CryptHashData where the algorithm is CALG_SHA_256 and the data being hashed is a password that is provided by the user. This password is not stored anywhere on the device.

Data on which crypto is applied

The data being encrypted is an XML file that is used to provision a WEH device. It contains information such as additional certificates to provision, what WLAN Profiles to apply, and what MDM server to connect with. WLAN profiles may contain credentials to an enterprise Wi-Fi access point. MDM server information contains the username and password that are used to enroll with the enterprise management server.

The data is generated and encrypted by the management server and transferred to the device, such as using an SD card. After the data is encrypted, it is base64 encoded. Once data is un-encrypted during WEH OOBE, it is stored as clear text on the MainOS partition in the enterprise shared data location.

Standards and protocols

The relevant standards that are used by the feature are AES encryption with a key length of 128, and 256-bit SHA hashing.

Crypto-related APIs

The following is a sequence of Crypto API calls made on the server to encrypt the provisioning file. The output of this sequence is then copied to the SD card.

Important note: The pbData parameter passed to CryptEncrypt should point to UTF-16 (WCHAR) characters and not UTF-8 ones.

```
CryptAcquireContext(&hProv, NULL, MS_ENH_RSA_AES_PROV, PROV_RSA_AES, CRYPT_VERIFYCONTEXT);
CryptCreateHash(hProv, CALG_SHA_256, 0, 0, &hHash);

// The data being hashed is a password provided by an IT administrator externally (e.g. when
using SCCM).
CryptHashData(hHash, pbKey, nKeySize, 0);
CryptDeriveKey(hProv, CALG_AES_128, hHash, 0, &hKey);
CryptDestroyHash(hHash);
CryptGetKeyParam(hKey, KP_BLOCKLEN, (BYTE*) &dwBlockLen, &dwLen, 0);
CryptEncrypt(hKey, 0, TRUE, 0, pData, &nInputSize, nBufferSize);
CryptDestroyKey(hKey);
CryptReleaseContext(hProv, 0);
Base64Encode(pbEncrypted, nEncryptedSize, (LPSTR)pbOutput, &nEncodedSize);
```

Similarly, the following sequence of Crypto API calls is used on the device to decrypt the provisioning file, which is encrypted by the server.

Important note: The pbData parameter passed to CryptEncrypt should point to UTF-16 (WCHAR) characters and not UTF-8 ones.

```
// The data is first base64 decoded.  pbInput is a pointer to the raw bytes of the
// encrypted file.
Base64Decode((LPCSTR)pbInput, nInputSize, pbDecoded, &nDecodedSize);

// Next, a sequence of CryptoAPIs.
CryptAcquireContext(&hProv, NULL, MS_ENH_RSA_AES_PROV, PROV_RSA_AES, CRYPT_VERIFYCONTEXT);
CryptCreateHash(hProv, CALG_SHA_256, 0, 0, &hHash);

// The data being hashed is a password provided via a Splash password box by the user.
CryptHashData(hHash, pbKey, nKeySize, 0);
CryptDeriveKey(hProv, CALG_AES_128, hHash, 0, &hKey);
CryptDestroyHash(hHash);
CryptDecrypt (hKey, 0, TRUE, 0, pData, &nPlainSize);
CryptDestroyKey(hKey);
CryptReleaseContext(hProv, 0);
```

Set time to sync automatically over Wi-Fi (Handheld 8.1)

If you are building a Wi-Fi only image, or if you plan to provision your devices without SIM cards, you cannot use the Network Identity and Time Zone (NITZ) method for setting and synching the time. NITZ is a common mechanism for setting time via a cellular network and is the standard way to set time on a Windows Phone, but it does not work without a cellular network. To set and sync the time on a device without a cellular network (Wi-Fi only image, or Wi-Fi only profile), you can use the built-in support for Network Time Protocol (NTP), which can be set by an NTP server as long as the device has a data (IP) connection. This connection can be either cellular or Wi-Fi data.

Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. NTP is intended to synchronize all participating computers to within a few milliseconds of Coordinated Universal Time (UTC). NTP can usually maintain time to within tens of milliseconds over the public Internet, and can achieve better than one millisecond of accuracy in local area networks under ideal conditions.

Note: NTP doesn't support Time Zone and Day Light savings information. Users will have to manually update the time zone.

To configure NTP in OOBE

To use NTP to automatically update the time on your device, you configure the NTP server and the Time Sync Interval. To do this, you add an "MCSF" characteristic to the prov.xml file.

```
<characteristic type="MCSF">
    <characteristic type="AutomaticTime">
      <parm name="NTPRegularSyncInterval"  value="1" datatype="integer" />
      <parm name="NTPServers" value=" time.contosso.com&#xF000;
time.windows.com&#xF000;&#xF000;" datatype="multiplestring" />
    </characteristic>
  </characteristic>
```

NTPRegularSyncInterval

The value in terms of hours and ranges from 1 to 168. The default value is 24.

NTPServers

The value is a multistring. The example shows  being used as a delimiter; NTPServer values should always end in   (double NULL).

Note: If no value is specified, the default value is time.windows.com. The strings shown here are examples only.

Enable near field communication (Handheld 8.1)

Near field communication (NFC) enables a Windows Embedded 8.1 Handheld powered device to communicate with an NFC tag or another NFC-enabled transmitting device. This section describes the components of an NFC tag that you should follow so that the tag works with your devices.

The NFC plug-in enables the administrator to provide a provisioning XML file during the out-of-box experience (OOBE) phase. The NFC plug-in allows an administrator to transfer provisioning information to persistent storage by tapping an unprovisioned Handheld 8.1 powered device to an NFC tag or NFC-enabled device. To use NFC for pre-provisioning a device, you must either prepare your own NFC tags by writing your provisioning XML file to a tag in the manner described in this section, or build the infrastructure needed to transmit a provisioning XML file between an NFC-enabled device and a Handheld 8.1 powered device during the OOBE phase.

Components of an NFC tag and an NFC-enabled device tag

This section describes the components of an NFC tag and an NFC-enabled device tag. Use an NFC tag for minimal provisioning and use an NFC-enabled device tag for larger provisioning XML files.

NFC tag components

NFC tags are suitable for very light applications where minimal provisioning is required. The size of NFC tags that contain provisioning XML files is typically 4 KB to 10 KB.

To write to an NFC tag, you will need to use an NFC Writer tool, or you can use the ProximityDevice class API to write your own custom tool to transfer your provisioning XML file to your NFC tag. The tool must publish a binary message (write) a Chunk data type to your NFC tag.

The following table describes the information that is required when writing to an NFC tag.

Required field	Description
Type	Windows.WEH.PreStageProv.Chunk The receiving device uses this information to understand information in the Data field.
Data	Tag data in UTF-8 format that has the Byte Order Mark (BOM) removed.

The following example shows how to write to an NFC tag. This example assumes that the tag is already in range of the writing device.

```
private void WriteProvXMLFileToTag(String provXMLFile)
{
    proximityDevice = Windows.Networking.Proximity.ProximityDevice.GetDefault();

    if (proximityDevice != null)
    {
```

```
        var dataWriter = new Windows.Storage.Streams.DataWriter();
        dataWriter.UnicodeEncoding = Windows.Storage.Streams.UnicodeEncoding.Utf8;
        dataWriter.WriteString(provXMLFile);
        var chunkPubId = proximityDevice.PublishBinaryMessage(
                    "Windows:WriteTag.WEH.PreStageProv.Chunk",
                    dataWriter.DetachBuffer());
    }
}
```

NFC-enabled device tag components

Provisioning from an NFC-enabled source device allows for larger provisioning XML files than can be transferred using an NFC tag. When provisioning from an NFC-enabled device, the total file size must not exceed 128 KB. Be aware that the larger the NFC file is, the longer it will take to transfer the provisioning file. Depending on your NFC hardware, the transfer time for a 128 KB file will vary between 2.5 seconds and 10 seconds.

To provision from an NFC-enabled source device, use ProximityDevice class API to write your own custom tool that transfers your provisioning XML file in chunks to your target Handheld 8.1 powered device. The tool must publish binary messages (transmit) a Header message, followed by one or more Chunk messages. The Header specifies the total amount of data that will be transferred to the target device; the Chunks must contain UTF-8 formatted provisioning data where the BOM is removed, as shown in the NFC tag components section.

The following table shows the header format.

Required field	Description
Type	Windows.WEH. PreStageProv.Header
Data	A string that is two UTF-8 semicolon delimited data-value pairs that identify the header version and the total number of bytes (1 through 131072). The following example shows the format: Vers=1.0;Len=<nnnn>

The target device caches the header information and then waits for the specified amount of data to arrive from the transmitter. The largest block of data that can be transmitted by NFC is 10 KB, so for large files the data must be transferred in separate chunks that are up to 10 KB in size. The receiver tallies the chunks and the content is reassembled. When all data has been received, the Handheld 8.1 powered device processes the provisioning data.

Although this method of transmitting and receiving protects against loss of data transmission, communication can be lost if the transmitting and receiving devices are out of range during the transmission or if the transmitter stops sending data. Communication between the two devices resynchronizes when a new header is transferred or when proximity is re-established.

The following example shows how to transmit a provisioning XML file to a target device. This example assumes that the devices are already in contact.

```
private void TransmitProvXMLFile(String provXMLFile)
{
    proximityDevice = Windows.Networking.Proximity.ProximityDevice.GetDefault();

    if (proximityDevice != null)
    {
```

```
        // Publish the header
        var dataWriter = new Windows.Storage.Streams.DataWriter();
        dataWriter.UnicodeEncoding = Windows.Storage.Streams.UnicodeEncoding.Utf8;
        dataWriter.WriteString("Vers=1.0;Len="+provXMLFile.Length.ToString()+";");
        proximityDevice.PublishBinaryMessage(
                "Windows.WEH.PreStageProv.Header",
                dataWriter.DetachBuffer());

        // Publish the data in chunks
        int maxMsgBytes = (int) proximityDevice.MaxMessageBytes;
        while (provXMLFile.Length > 0)
        {
            // Determine the maximum amount of data to send.
            int transmitSize=Math.Min(provXMLFile.Length,maxMsgBytes);

            // Prepare the chunk for transmission to peer device
            String fileChunk = provXMLFile.Substring(0, transmitSize);
            dataWriter = new Windows.Storage.Streams.DataWriter();
            dataWriter.UnicodeEncoding = Windows.Storage.Streams.UnicodeEncoding.Utf8;
            dataWriter.WriteString(fileChunk);

            // Publish chunk to peer
            proximityDevice.PublishBinaryMessage(
                    "Windows.WEH.PreStageProv.Chunk",
                    dataWriter.DetachBuffer());

            // Reduce the source data
            provXMLFile =  provXMLFile.Remove(0, transmitSize);
        }
    }
```

For more information about the ProximityDevice class API, see ProximityDevice class on the Developer Center.

Enable or disable NFC capabilities

The administrator can control whether to enable NFC capabilities on the device. When NFC is allowed, the user can change several settings by using Settings/NFC. When disabled, a message appears on the Settings page that NFC is disabled by company policy and the user cannot change the settings.

To disable NFC, set AllowNFC to 0; otherwise, set it to 1.

The following example shows how to allow an NFC tag.

```
<Add>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/My/Connectivity/AllowNFC</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>1</Data>
   </Item>
</Add>
```

The following example shows how to disallow an NFC tag.

```
<Add>
   <CmdID>3</CmdID>
   <Item>
```

```
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/My/Connectivity/AllowNFC</LocURI>
      </Target>
      <Meta>
         <Format xmlns="syncml:metinf">int</Format>
      </Meta>
      <Data>0</Data>
   </Item>
</Add>
```

The following example shows how to query the current policy value.

```
<Get>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/PolicyManager/Device/Connectivity/AllowNFC</LocURI>
      </Target>
   </Item>
</Get>
```

EnableDeviceEnrollment Request and Response (Handheld 8.1)

The differences from Windows Phone when bulk enrollment is enabled by setting EnableDeviceEnrollment to true are as follows:

- The discovery service is skipped.
- The get certificate policy is skipped.
- There is a modified certificate enrollment web service request for a specific Server URI.

For more information about EnableDeviceEnrollment, see The provisioning XML file (Prov.xml) and EnterpriseExt configuration service provider topics for Handheld 8.1.

Modified certificate enrollment web service request

The SOAP request is identical to Windows Phone 8.1 Enterprise Device Management Protocol v2.0 SOAP request except for the following:

1. ac.AdditionalContext/ac:ContextItem[@Name='DeviceType']/ac:Value" is WindowsEmbeddedHandheld.
2. "ac.AdditionalContext/ac:ContextItem[@Name='DeviceId']/ac:Value" is new and represents a unique Device Identifier that persists across resets.
3. The modified certificate request has a DeviceID value embedded, as described here.

Note: DeviceID is referenced throughout this document. Each device running Handheld 8.1 must have a unique DeviceID that can be used to identify the specific device running Handheld 8.1 and only that device. The DeviceID should remain constant throughout the lifetime of the device running Handheld 8.1 and survive any form of reset.

SOAP Request

The following sample RST message shows a SOAP request.

Header

```
POST /EnrollmentServer/DeviceEnrollmentWebService.svc HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
User-Agent: Windows Phone 8 Enrollment Client
Host: enrolltest.contoso.com
Cache-Control: no-cache

<s:Envelope
    xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
    xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"
    xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"
    xmlns:ac="http://schemas.xmlsoap.org/ws/2006/12/authorization"
>
    <s:Header>
        <a:Action

s:mustUnderstand="1">http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RST/wstep
        </a:Action>
        <a:MessageID>urn:uuid:d7cfb6f6-9bf4-4771-bdc2-d4bf66f8b4f5</a:MessageID>
        <a:ReplyTo>
            <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
        </a:ReplyTo>
        <a:To

s:mustUnderstand="1">https://enrolltest.contoso.com/ENROLLMENTSERVICE/DeviceEnrollmentService.
svc
        </a:To>
        <o:Security
            s:mustUnderstand="1"
            xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <o:UsernameToken
                u:Id="uuid-9c2e505d-1795-46db-a520-7f69fec0aaa3-3">
                <o:Username>user@contoso.com</o:Username>
                <o:Password
                    Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">
                    mypassword
                </o:Password>
            </o:UsernameToken>
        </o:Security>
    </s:Header>
    <s:Body>
        <wst:RequestSecurityToken>
            <wst:TokenType>
http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
            </wst:TokenType>
            <wst:RequestType>
http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue
            </wst:RequestType>
            <wsse:BinarySecurityToken
                ValueType="http://schemas.microsoft.com/windows/pki/2009/01/enrollment#PKCS10"
                EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd#base64binary">
DER format PKCS#10 modified certificate request in Base64 encoding Inserted Here
            </wsse:BinarySecurityToken>
            <ac:AdditionalContext>
```

```
                <ac:ContextItem Name="DeviceType">
                    <ac:Value>WindowsEmbeddedHandheld</ac:Value>
                </ac:ContextItem>
                <ac:ContextItem Name="ApplicationVersion">
                    <ac:Value>8.1.0.0</ac:Value>
                </ac:ContextItem>
                <ac:ContextItem Name="DeviceId">
                    <ac:Value>bdd19ca0-dfce-497a-bea1-fda234e52b36</ac:Value>
                </ac:ContextItem>
            </ac:AdditionalContext>
        </wst:RequestSecurityToken>
    </s:Body>
</s:Envelope>
```

X509 certificate request

The primary function of enrollment is for the enrollment service to fulfill an X509 Certificate request on behalf of the client. The X509 Certificate request is the same format as Windows Mobile 6.5 with two exceptions. Handheld 8.1 device certificate requests must provide the following:

- The unique DeviceID as the SUBJECT Name in the request (this is a fixed value in Windows Phone 8.1).
- The unique DeviceID as a custom extension in the request.

SUBJECT Name

Providing the DeviceID in the subject name gives a certification authority (CA) administrator leeway in how they want to organize certificates issued to devices running Handheld 8.1. If the SUBJECT Name is guaranteed to be unique, the CA administrator may choose to use the SUBJECT Name from the request when issuing the certificate.

Custom Extension

Providing the DeviceID as a custom extension also gives the CA administrator the ability to assign the SUBJECT Name from Active Directory Domain Services (AD DS), essentially using the bulk enrollment account username, instead of the SUBJECT Name from the request, which is less secure. By having the DeviceID in a custom extension, the CA administrator can still identify the management certificate for a specific device running Handheld 8.1.

The object identifier (also known as OID) used for the custom extension must be: "1.3.6.1.4.1.311.66.1.0"

The first seven parts are the Microsoft standard object identifier. The 66.1.0 is made up and does not collide with the known object identifiers found at http://support.microsoft.com/kb/287547.

SOAP response

The SOAP response returns the fulfilled X509 Certificate for the client. The following sample shows a complete SOAP response from the service.

```
<s:Envelope
    xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing"
    xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
    <s:Header>
        <a:Action
```

```
s:mustUnderstand="1">http://schemas.microsoft.com/windows/pki/2009/01/enrollment/RSTRC/wstep
        </a:Action>
        <a:RelatesTo>urn:uuid:66cddf7d-0227-48c7-a270-935fdfd27359</a:RelatesTo>
        <o:Security
            s:mustUnderstand="1"
            xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <u:Timestamp u:Id="_0">
                <u:Created>2013-08-27T15:32:33.256Z</u:Created>
                <u:Expires>2013-08-27T15:37:33.256Z</u:Expires>
            </u:Timestamp>
        </o:Security>
    </s:Header>
    <s:Body>
        <RequestSecurityTokenResponseCollection
            xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
            <RequestSecurityTokenResponse>
                <TokenType>

http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollmentToken
                </TokenType>
                <RequestedSecurityToken>
                    <BinarySecurityToken

ValueType="http://schemas.microsoft.com/5.0.0.0/ConfigurationManager/Enrollment/DeviceEnrollme
ntProvisionDoc"
                        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd#base64binary"
                        xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
                        Binary Encoded Enrollment Response
                    </BinarySecurityToken>
                </RequestedSecurityToken>
                <RequestID
xmlns="http://schemas.microsoft.com/windows/pki/2009/01/enrollment">0</RequestID>
            </RequestSecurityTokenResponse>
        </RequestSecurityTokenResponseCollection>
    </s:Body>
</s:Envelope>
```

Enrollment response

The enrollment response is base-64 encoded in the SOAP message. The following listing shows the decoded enrollment response.

```
<wap-provisioningdoc version="1.1">
  <characteristic type="CertificateStore">
    <characteristic type="Root">
      <characteristic type="System">
        <characteristic type="E22790C0148DDF3B699C5706B7881FDED60B51EB">
          <parm name="EncodedCertificate" value="CERTIFICATE GOES HERE" />
        </characteristic>
      </characteristic>
    </characteristic>
    <characteristic type="My">
      <characteristic type="User">
        <characteristic type="A9D27BAA6DB9EBE54F0750494A1ABD323281A0F6">
          <parm name="EncodedCertificate" value="CERTIFICATE GOES HERE"
          <characteristic type="PrivateKeyContainer">
            <parm name="KeySpec" value="2" />
            <parm name="ContainerName" value="ConfigMgrEnrollment" />
```

```
            <parm name="ProviderType" value="1" />
          </characteristic>
        </characteristic>
      </characteristic>
    </characteristic>
  </characteristic>
  <characteristic type="APPLICATION">
    <parm name="APPID" value="w7" />
    <parm name="PROVIDER-ID" value="SCConfigMgr" />
    <parm name="NAME" value="Microsoft" />
    <parm name="ADDR"
value="https://enrolltest.contoso.com:443/omadm/WindowsPhoneEmbedded.ashx" />
    <parm name="CRLCheck" value="0" />
    <parm name="CONNRETRYFREQ" value="6" />
    <parm name="INITIALBACKOFFTIME" value="30000" />
    <parm name="MAXBACKOFFTIME" value="120000" />
    <parm name="BACKCOMPATRETRYDISABLED" />
    <parm name="DEFAULTENCODING" value="application/vnd.syncml.dm+wbxml" />
    <parm name="SSLCLIENTCERTSEARCHCRITERIA"
value="Subject=DC%3dCOM%2cDC%3dCONTOSO%2cDC%3dENROLLTEST%2cCN%3dUsers%2cCN%3duser%40contoso.co
m&amp;Stores=MY%5CUser" />
    <characteristic type="APPAUTH">
      <parm name="AAUTHLEVEL" value="CLIENT" />
      <parm name="AAUTHTYPE" value="DIGEST" />
      <parm name="AAUTHSECRET" value="dummy" />
      <parm name="AAUTHDATA" value="nonce" />
    </characteristic>
    <characteristic type="APPAUTH">
      <parm name="AAUTHLEVEL" value="APPSRV" />
      <parm name="AAUTHTYPE" value="DIGEST" />
      <parm name="AAUTHNAME" value="dummy" />
      <parm name="AAUTHSECRET" value="dummy" />
      <parm name="AAUTHDATA" value="nonce" />
    </characteristic>
  </characteristic>
  <characteristic type="Registry">
    <characteristic type="HKLM\Software\Microsoft\Enrollment">
      <parm name="RenewalPeriod" value="42" datatype="integer" />
    </characteristic>
    <characteristic type="HKLM\Software\Microsoft\Enrollment\OmaDmRetry">
      <parm name="NumRetries" value="8" datatype="integer" />
      <parm name="RetryInterval" value="15" datatype="integer" />
      <parm name="AuxNumRetries" value="5" datatype="integer" />
      <parm name="AuxRetryInterval" value="3" datatype="integer" />
      <parm name="Aux2NumRetries" value="0" datatype="integer" />
      <parm name="Aux2RetryInterval" value="480" datatype="integer" />
    </characteristic>
  </characteristic>
  <characteristic type="DMClient">
    <characteristic type="Provider">
      <characteristic type="SCConfigMgr">
        <parm name="EntDeviceName" value="Bulk Profile_WindowsEmbeddedHandheld_1_e7a3c90f-c"
datatype="string" />
      </characteristic>
    </characteristic>
  </characteristic>
  <characteristic type="EnterpriseAppManagement">
    <characteristic type="4000000001">
      <parm datatype="string" name="EnrollmentToken" value="ENROLLMENT TOKEN GOES HERE"
      <parm datatype="string" name="StoreProductId" value="{00000000-0000-0000-0000-
000000000000}" />
      <parm datatype="string" name="StoreURI" value="" />
```

```
      <parm datatype="string" name="StoreName" value="Contoso App Store" />
      <parm datatype="string" name="CertificateSearchCriteria" value="
DC%3dCOM%2cDC%3dCONTOSO%2cDC%3dENROLLTEST%2cCN%3dUsers%2cCN%3duser%40contoso.com" />
      <parm datatype="string" name="CRLCheck" value="0" />
    </characteristic>
  </characteristic>
</wap-provisioningdoc>
```

Apps Corner (Handheld 8.1)

~~There are two ways to~~You can export ~~generate~~ a WEHLockdown.xml file to an SD card in Apps Corner and then use it provision devices~~that you can provision to devices.~~

- ~~Pre-installed an assigned access version.~~
- ~~Export an XML file to an SD card in Apps Corner~~

~~For more information about pre-installed assigned access version and lockdown XML, see the Administrator Guide for Windows Embedded 8.1 Handheld. This topic describes the procedure for exporting an XML file to an SD card.~~

Steps for exporting a file in Apps Corner:

1. Manually setup one device. Make sure there is an SD card.
2. Install the desired enterprise app on the device,
3. In the control panel, click Settings > Advanced > Apps Corner > Export the configuration to the SD card.

This generates a WEHLockdown XML file that you ~~need~~ can ~~to~~ install on every device.

Assigned Access (Handheld 8.1)

Formatted: Heading 2,h2

Enabling the Assigned Access feature requires the addition of special registry keys and placing the WEHLockdown.xml onto the device in a specific folder of the file structure. The XML file must be authored by the enterprise following the guidelines defined in the Administrator Guide for Windows Embedded 8.1 Handheld. The file must be named WEHLockdown.xml and placed into the \Data\SharedData\Enterprise\Persistent\ directory.

The following table shows the relevant reqistry keys for Assigned Access.

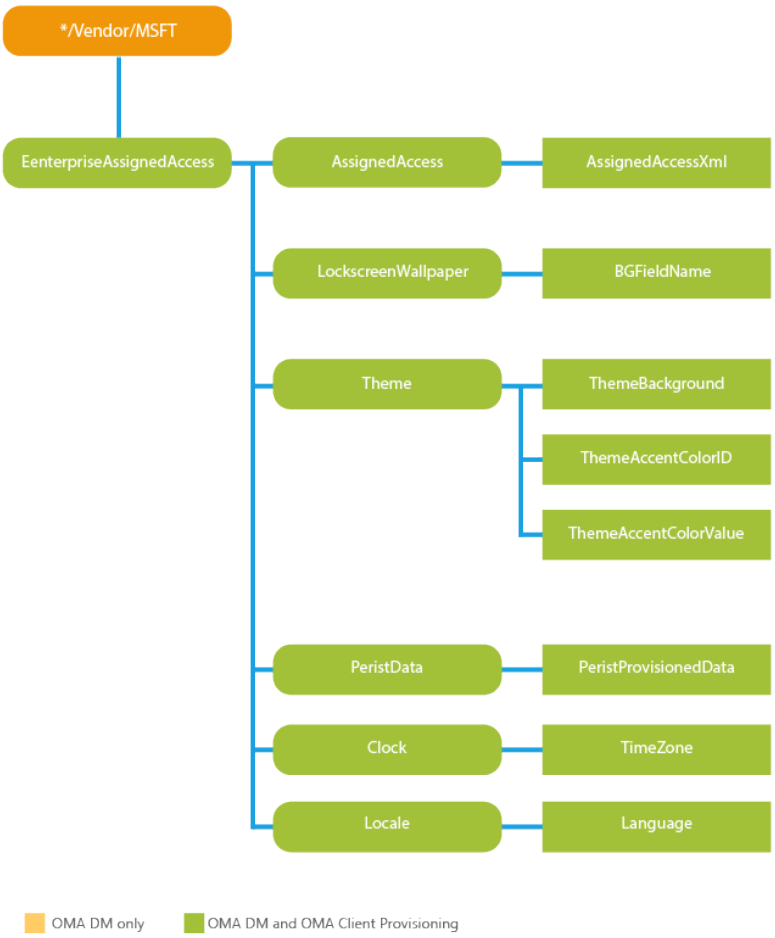
Registry key	Value	Value data
HKLM\System\Features\	Lockdown	WindowsEmbeddedDeviceLockdownProfile.dll
HKLM\System\Features\	ButtonRemapping	WEHButtonRouter.dll

EnterpriseAssignedAccess configuration service provider (Handheld 8.1)

The EnterpriseAssignedAccess configuration service provider allows Information Technology (IT) administrators to configure settings, such as language and themes, lock down a device, and use Windows Embedded 8.1 Handheld features to configure custom layouts on a device. For example, the administrator can lock down a device so that only applications specified in an Allow list are available. Apps not on the Allow list remain installed on the device, but are hidden from view.

Important note: This CSP applies only to Windows Embedded 8.1 Handheld devices.

The following image shows the EnterpriseAssignedAccess configuration service provider in tree format as used by both the Open Mobile Alliance (OMA) Device Management (DM) and OMA Client Provisioning.



.Vendor/MSFT/EnterpriseAssignedAccess/

The root node for the EnterpriseAssignedAccess configuration service provider. Supported operations: Add, Replace, and Get.

AssignedAccess/

The parent node of assigned access XML.

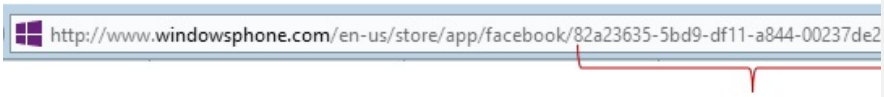
AssignedAccess/AssignedAccessXml

The XML code that controls the assigned access settings that will be applied to the device.

Supported operations: Add, Replace, and Get.

The Apps and Settings sections of Prov.xml constitute an Allow list. Any app or setting that is not specified in AssignedAccessXML will not be available on the device to users. The following table describes the entries in Prov.xml.

Important note: The formatting used in the examples in this table cannot be used in your actual provisioning file. The example is provided in this format for readability only. The provisioning file must use escaped characters for EnterpriseAssignedAcces, such as < instead of < because xml is embedded in xml. Do not replace the escaped characters in the provisioning file. See The provisioning XML file (Prov.xml) for the correct formatting.

Entry	Description
ActionCenter	You can enable or disable the Action Center (formerly known as Notification Center) on the device. Set to true to enable the Action Center, or set to false to disable the Action Center. Example: `<ActionCenter enabled="true"></ActionCenter>`
StartScreenSize	Specify the size of the Start screen. Large sets the width to be big enough to hold six small tiles, or the equivalent. For example, six small tiles are about the same as one large tile and one medium tile. Small sets the width to 4, which is equivalent to the total width of four small tiles. Example: `<StartScreenSize>Large</StartScreenSize>`
Application	Provide the product ID for each app that will be available on the device. To obtain the product ID for apps that you install from the Windows Phone Store, open a browser and navigate to the installation page for the app. In the URL, you will see the GUID for the app, as shown in the following illustration.  You can find the product ID for a locally developed app in the AppManifest.xml file of the app. Include PinToStart to display an app on the Start screen. For apps pinned to the Start screen, identify a tile size (small, medium, or large), and a location. The size of a small tile is 1 column x 1 row, a medium tile is 2 x 2, and a large tile is 4 x 2. For the tile location, the first value indicates the column and the second value indicates the row. A value of 0 indicates the first column, a value of 1 indicates the second column, and so on. Include autoRun as an attribute to configure the application to run automatically. Example: `<Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}" autoRun="true">`

	<pre><PinToStart>
	<Size>Large</Size>
	<Location>
	<LocationX>0</LocationX>
	<LocationY>2</LocationY>
	</Location>
	</PinToStart>
	</Application></pre>

AppInstall	Provide the installation type ("SDCard" or "Network"), product ID, application file path, and license file path to install an application during the OOBE provisioning. Applications can be installed during OOBE from an SD card or from a shared network location.
	
	Example:
	<pre><characteristic type="AppInstall">
	<characteristic type="SDCard">
	<parm name="ProductID" value="{912627c8-174c-4a49-ac53-a2b8e4a5be37}"/>
	<parm name="AppXPath" value="Appx\ReliabilityAppxV1.appx"/>
	<parm name="LicensePath" value="Appx\ReliabilityAppxV1_license.xml"/>
	</characteristic>
	<characteristic type="Network">
	<parm name="ProductID" value="{912627c8-174c-4a49-ac53-a2b8e4a5be37}"/>
	<parm name="AppXPath" value="\\SharedFolder\ReliabilityAppxV1.appx"/>
	<parm name="LicensePath"
	value="\\SharedFolder\ReliabilityAppxV1_license.xml"/>
	</characteristic>
	</characteristic></pre>
	The following example shows how to use an XAP package:
	<pre><characteristic type="AppInstall">
	<characteristic type="SDCard">
	<parm name="ProductID" value="{F8240AA8-B1C7-4a9c-8914-79BA6A466475}"/>
	<parm name="XAPPath" value="MEGSLTestGame.xap"/>
	<parm name="LicensePath" value="MEGSLTestGame_License.xml"/>
	</characteristic>
	</characteristic></pre>

Settings	Provide the setting name that will be available on the device.
	
	Example:
	<pre><Settings>
	<System name="Microsoft.Themes" />
	<Application name="Microsoft.Search" />
	</Settings></pre>
	Important note: If the Microsoft.DateTime setting is not locked down, users can change the time on the device. This can cause scheduled maintenance and communication with the MDM server to occur at the wrong time.

Buttons	The following list identifies the hardware buttons on the device that you can lock down in ButtonLockdownList. When a user taps a button that is in the lockdown list, nothing will happen. • Start (**Note:** Lock down of the Start button only prevents the press and hold event.)

- Back
- Search
- Camera
- Custom1
- Custom2
- Custom3

Note: Custom buttons are hardware buttons that can be added to devices by OEMs.

Example:

```
<Buttons>
   <ButtonLockdownList>
      <!-- Lockdown all buttons -->
         <Button name="Search">
         </Button>
         <Button name="Camera">
         </Button>
         <Button name="Custom1">
         </Button>
         <Button name="Custom2">
         </Button>
         <Button name="Custom3">
         </Button>
   </ButtonLockdownList>
```

The Search and custom buttons can be remapped or configured to open a specific application. Button remapping takes effect for the device and applies to all users.

Note: The lockdown settings for a button, per user role, will apply regardless of the button mapping.

Caution: Button remapping can enable a user to open an application that is not in the Allow list. Use button lock down to prevent application access for a user role.

To remap a button in Prov.xml, you supply the button name, the button event (typically "press"), and the product ID for the application the button will open.

Example:

```
<ButtonRemapList>
   <Button name="Search">
      <ButtonEvent name="Press">
         <!-- Alarms -->
         <Application productId="{08179793-ED2E-45EA-BA12-BDE3EE9C3CE3}"
parameters="" />
          </ButtonEvent>
   </Button>
</ButtonRemapList>
```

Disabling navigation buttons

To disable navigation buttons (such as Home or Back) in prov.xml, you supply the name (for example, Start) and button event (typically "press").

The following section contains a sample WEHLockdown.xml file that shows how to disable navigation buttons.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
    <Default>
        <ActionCenter enabled="false" />
        <Apps>
            <!-- Settings -->
            <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}">
                <PinToStart>
                    <Size>Large</Size>
                    <Location>
                        <LocationX>0</LocationX>
                        <LocationY>0</LocationY>
                    </Location>
                </PinToStart>
            </Application>

            <!-- Phone Apps -->
            <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5611}">
                <PinToStart>
                    <Size>Small</Size>
                    <Location>
                        <LocationX>2</LocationX>
                        <LocationY>2</LocationY>
                    </Location>
                </PinToStart>
            </Application>
        </Apps>
        <Buttons>
            <ButtonLockdownList>
                <Button name="Start">
                    <ButtonEvent name="Press" />
                </Button>
                <Button name="Back">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Search">
                    <ButtonEvent name="All" />
                </Button>
                <Button name="Camera">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Custom1">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Custom2">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Custom3">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
            </ButtonLockdownList>
            <ButtonRemapList />
```

	<pre> </Buttons> <MenuItems> <DisableMenuItems/> </MenuItems> <Settings> <System name="Microsoft.About" /> <System name="Microsoft.FlashAppSetting" /> <System name="Microsoft.CompanyAccount" /> <System name="Microsoft.WiFi" /> <Application name="Microsoft.Search" /> <Application name="Microsoft.IE" /> <Application name="Microsoft.Maps" /> <Application name="Microsoft.Messaging" /> <Application name="Microsoft.OfficeMobile" /> <Application name="Microsoft.Contacts" /> <Application name="Microsoft.Phone" /> </Settings> <Tiles> <EnableTileManipulation/> </Tiles> <StartScreenSize>Small</StartScreenSize> </Default> </HandheldLockdown></pre>

MenuItems	Use DisableMenuItems to prevent use of the context menu, which is displayed when a user presses and holds an application in the All Programs list. You can include this entry in the default profile and in any additional user role profiles that you create.
	
	Example:
	
	<pre><MenuItems>
	   <DisableMenuItems/>
	</MenuItems></pre>
	Important note: If DisableMenuItems is not included in a profile, users of that profile can uninstall apps.

Tiles	**Turning-on tile manipulation** By default, under Assigned Access, tile manipulation is turned off (blocked) and only available if enabled in the user's profile. If tile manipulation is enabled in the user's profile, they can pin/unpin, move, and resize tiles based on their preferences. When multiple people use one device and you want to enable tile manipulation for multiple users, you must enable it for each user in their user profile. **Important note:** If a device is turned off then back on, the tiles reset to their predefined layout. If a device has only one profile, the only way to reset the tiles is to turn off then turn on the device. If a device has multiple profiles, the device resets the tiles to the predefined layout based on the logged-in user's profile. The following sample file contains code for enabling tile manipulation. Note: Tile manipulation is disabled when you don't have a <Tiles> node in WEHLockdown.xml, or if you have a <Tiles> node but don't have the <EnableTileManipulation/> node.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HandheldLockdown version="1.0" >
    <Default>
        <ActionCenter enabled="false" />
        <Apps>
            <!-- Settings -->
            <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5601}">
                <PinToStart>
                    <Size>Large</Size>
                    <Location>
                        <LocationX>0</LocationX>
                        <LocationY>0</LocationY>
                    </Location>
                </PinToStart>
            </Application>

            <!-- Phone Apps -->
            <Application productId="{5B04B775-356B-4AA0-AAF8-6491FFEA5611}">
                <PinToStart>
                    <Size>Small</Size>
                    <Location>
                        <LocationX>2</LocationX>
                        <LocationY>2</LocationY>
                    </Location>
                </PinToStart>
            </Application>
        </Apps>
        <Buttons>
            <ButtonLockdownList>
                <Button name="Start">
                    <ButtonEvent name="Press" />
                </Button>
                <Button name="Back">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Search">
                    <ButtonEvent name="All" />
                </Button>
                <Button name="Camera">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Custom1">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Custom2">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
                <Button name="Custom3">
                    <ButtonEvent name="Press" />
                    <ButtonEvent name="PressAndHold" />
                </Button>
            </ButtonLockdownList>
            <ButtonRemapList />
        </Buttons>
        <MenuItems>
```

```
            <DisableMenuItems/>
        </MenuItems>
        <Settings>
            <System name="Microsoft.About" />
            <System name="Microsoft.FlashAppSetting" />
            <System name="Microsoft.CompanyAccount" />
            <System name="Microsoft.WiFi" />
            <Application name="Microsoft.Search" />
            <Application name="Microsoft.IE" />
            <Application name="Microsoft.Maps" />
            <Application name="Microsoft.Messaging" />
            <Application name="Microsoft.OfficeMobile" />
            <Application name="Microsoft.Contacts" />
            <Application name="Microsoft.Phone" />
        </Settings>
        <Tiles>
            <EnableTileManipulation/>
        </Tiles>
        <StartScreenSize>Small</StartScreenSize>
    </Default>
</HandheldLockdown>
```

LockscreenWallpaper/

The parent node of the lock screen-related parameters that let adminstrators query and manage the lock screen image on devices. Supported operations: Add, Replace, and Get.

LockscreenWallpaper/BGFileName

The file name of the lock screen. The image file for the lock screen can be in .jpg or .png format and must not exceed 2 MB. The file name can also be in the Universal Naming Convention (UNC) format, in which case the device downloads it from the shared network and then sets it as the lock screen wallpaper.

Supported operations: Add, Replace, and Get.

Theme/

The parent node of theme-related parameters.

Theme/ThemeBackground

Indicates whether the background color is light or dark. Set to 0 for light; set to 1 for dark.

Theme/ThemeAccentColorID

The accent color to apply as the foreground color for tiles, controls, and other visual elements on the device. The following table shows the possible values.

Value	Description
0	Lime
1	Green
2	Emerald
3	Teal (Viridian)
4	Cyan (Blue)
5	Cobalt
6	Indigo
7	Violet (Purple)

8	Pink
9	Magenta
10	Crimson
11	Red
12	Orange (Mango)
13	Amber
14	Yellow
15	Brown
16	Olive
17	Steel
18	Mauve
19	Sienna
101 through 104	Optional colors, as defined by the OEM
151	Custom accent color for Enterprise

For more information about accent colors, see Themes for Windows Phone.

Supported operations: Add, Replace, and Get.

Theme/ThemeAccentColorValue

A 6-character string for the accent color to apply to controls and other visual elements.

To use a custom accent color for Enterprise, enter 151 for ThemeAccentColorID before ThemeAccentColorValue in Prov.xml. ThemeAccentColorValue configures the custom accent color using hex values for red, green, and blue, in RRGGBB format. For example, enter FF0000 for red.

PersistData

The parent node of whether to persist data that has been provisioned on the device.

PersistData/PersistProvisionedData

Indicates whether to retain provisioned data when the user resets a device. Set to 0 if you do not want to persist provisioned data; set to 1 to persist it.

Note: PersistProvisionedData works with the RemoteWipe configuration service provider on Windows Phone OS. When executed, PersistProvisionedData backs up the persistent store folder so that the RemoteWipe configuration service provider can wipe the device. The information that was backed up is restored to the device when it resumes.

Clock/TimeZone

A string that specifies the time zone of the device. The following table shows the possible values.

Value	Time zone
0	UTC-12 International Date Line West
100	UTC+13 Samoa
110	UTC-11 Coordinated Universal Time-11
200	UTC-10 Hawaii
300	UTC-09 Alaska
400	UTC-08 Pacific Time (US & Canada)
410	UTC-08 Baja California
500	UTC-07 Mountain Time (US & Canada)

510	UTC-07 Chihuahua, La Paz, Mazatlan
520	UTC-07 Arizona
600	UTC-06 Saskatchewan
610	UTC-06 Central America
620	UTC-06 Central Time (US & Canada)
630	UTC-06 Guadalajara, Mexico City, Monterrey
700	UTC-05 Eastern Time (US & Canada)
710	UTC-05 Bogota, Lima, Quito
720	UTC-05 Indiana (East)
800	UTC-04 Atlantic Time (Canada)
810	UTC-04 Cuiaba
820	UTC-04 Santiago
830	UTC-04 Georgetown, La Paz, Manaus, San Juan
840	UTC-04 Caracas
850	UTC-04 Asuncion
900	UTC-03:30 Newfoundland
910	UTC-03 Brasilia
920	UTC-03 Greenland
930	UTC-03 Montevideo
940	UTC-03 Cayenne, Fortaleza
950	UTC-03 Buenos Aires
960	UTC-03 Salvador
1000	UTC-02 Mid-Atlantic
1010	UTC-02 Coordinated Universal Time-02
1100	UTC-01 Azores
1110	UTC-01 Cape Verde Is.
1200	UTC Dublin, Edinburgh, Lisbon, London
1210	UTC Monrovia, Reykjavik
1220	UTC Casablanca
1230	UTC Coordinated Universal Time
1300	UTC+01 Belgrade, Bratislava, Budapest, Ljubljana, Prague
1310	UTC+01 Sarajevo, Skopje, Warsaw, Zagreb
1320	UTC+01 Brussels, Copenhagen, Madrid, Paris
1330	UTC+01 West Central Africa
1340	UTC+01 Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
1350	UTC+01 Windhoek
1360	UTC+01 Tripoli
1400	UTC+02 E. Europe
1410	UTC+02 Cairo
1420	UTC+02 Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
1430	UTC+02 Athens, Bucharest
1440	UTC+02 Jerusalem
1450	UTC+02 Amman
1460	UTC+02 Beirut
1470	UTC+02 Harare, Pretoria
1480	UTC+02 Damascus

1490	UTC+02 Istanbul
1500	UTC+03 Kuwait, Riyadh
1510	UTC+03 Baghdad
1520	UTC+03 Nairobi
1530	UTC+03 Kaliningrad, Minsk
1540	UTC+04 Moscow, St. Petersburg, Volgograd
1550	UTC+03 Tehran
1600	UTC+04 Abu Dhabi, Musca
1610	UTC+04 Baku
1620	UTC+04 Yerevan
1630	UTC+04 Kabul
1640	UTC+04 Tbilisi
1650	UTC+04 Port Louis
1700	UTC+06 Ekaterinburg
1710	UTC+05 Tashkent
1720	UTC+05 Chennai, Kolkata, Mumbai, New Delhi
1730	UTC+05 Sri Jayawardenepura
1740	UTC+05 Kathmandu
1750	UTC+05 Islamabad, Karachi
1800	UTC+06 Astana
1810	UTC+07 Novosibirsk
1820	UTC+06 Yangon (Rangoon)
1830	UTC+06 Dhaka
1900	UTC+08 Krasnoyarsk
2000	UTC+08 Beijing, Chongqing, Hong Kong, Urumqi
2010	UTC+09 Irkutsk
2020	UTC+08 Kuala Lumpur, Singapore
2030	UTC+08 Taipei
2040	UTC+08 Perth
2050	UTC+08 Ulaanbaatar
2100	UTC+09 Seoul
2110	UTC+09 Osaka, Sapporo, Tokyo
2120	UTC+10 Yakutsk
2130	UTC+09 Darwin
2140	UTC+09 Adelaide
2200	UTC+10 Canberra, Melbourne, Sydney
2210	UTC+10 Brisbane
2220	UTC+10 Hobart
2230	UTC+11 Vladivostok
2240	UTC+10 Guam, Port Moresby
2300	UTC+11 Solomon Is., New Caledonia
2310	UTC+12 Magadan
2400	UTC+12 Fiji
2410	UTC+12 Auckland, Wellington
2420	UTC+12 Petropavlovsk-Kamchatsky
2430	UTC+12 Coordinated Universal Time +12
2500	UTC+13 Nuku'alofa

Locale/Language/

The culture code that identifies the language to display on a device, and specifies the formatting of numbers, currencies, time, and dates. For language values, see Locale IDs Assigned by Microsoft.

The language setting is configured in the Default User profile only.

OMA client provisioning examples

The XML examples in this section show how to perform various tasks by using OMA client provisioning.

Note: These examples are XML snippets and do not include all sections that are required for a complete Prov.xml file.

Assigned Access settings

The following example shows how to add a new policy.

```
<wap-provisioningdoc>
  <characteristic type="EnterpriseAssignedAccess">
    <characteristic type="AssignedAccess">
      <parm name=" AssignedAccessXml" datatype="string"
            value="&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-
8&quot;?&gt;&lt;HandheldLockdown
version=&quot;1.0&quot;&gt;&lt;Default&gt;&lt;Apps&gt;&lt;Application
productId=&quot;{5B04B775-356B-4AA0-AAF8-6491FFEA5615}&quot;
pinToStart=&quot;1&quot;/&gt;&lt;Application productId=&quot;{5B04B775-356B-4AA0-AAF8-
6491FFEA5612}&quot; pinToStart=&quot;0&quot;/&gt;&lt;/Apps&gt;&lt;Settings&gt;&lt;System
name=&quot;Microsoft.Themes&quot; /&gt;&lt;System name=&quot;Microsoft.About&quot;
/&gt;&lt;/Settings&gt;&lt;Buttons&gt;&lt;ButtonLockdownList&gt;&lt;Button
name=&quot;Start&quot;&gt;&lt;ButtonEvent name=&quot;Press&quot; /&gt;&lt;ButtonEvent
name=&quot;PressAndHold&quot; /&gt;&lt;/Button&gt;&lt;Button
name=&quot;Camera&quot;&gt;&lt;ButtonEvent name=&quot;Press&quot; /&gt;&lt;ButtonEvent
name=&quot;PressAndHold&quot; /&gt;&lt;/Button&gt;&lt;Button
name=&quot;Search&quot;&gt;&lt;ButtonEvent name=&quot;Press&quot; /&gt;&lt;ButtonEvent
name=&quot;PressAndHold&quot;
/&gt;&lt;/Button&gt;&lt;/ButtonLockdownList&gt;&lt;ButtonRemapList/&gt;&lt;/Buttons&gt;&lt;Men
uItems&gt;&lt;DisableMenuItems/&gt;&lt;/MenuItems&gt;&lt;/Default&gt;&lt;RoleList&gt;&lt;Role
guid=&quot;{76C01983-A872-4C4E-B4C6-321EAC709CEA}&quot;
name=&quot;Associate&quot;&gt;&lt;Apps&gt;&lt;Application productId=&quot;{5B04B775-356B-4AA0-
AAF8-6491FFEA5615}&quot; pinToStart=&quot;1&quot;/&gt;&lt;/Apps&gt;&lt;Settings&gt;&lt;System
name=&quot;Microsoft.Themes&quot; /&gt;&lt;System name=&quot;Microsoft.About&quot;
/&gt;&lt;/Settings&gt;&lt;Buttons&gt;&lt;ButtonLockdownList&gt;&lt;Button
name=&quot;Start&quot;&gt;&lt;ButtonEvent name=&quot;Press&quot; /&gt;&lt;ButtonEvent
name=&quot;PressAndHold&quot; /&gt;&lt;/Button&gt;&lt;Button
name=&quot;Camera&quot;&gt;&lt;ButtonEvent name=&quot;Press&quot; /&gt;&lt;ButtonEvent
name=&quot;PressAndHold&quot;
/&gt;&lt;/Button&gt;&lt;/ButtonLockdownList&gt;&lt;ButtonRemapList/&gt;&lt;/Buttons&gt;&lt;Men
uItems&gt;&lt;DisableMenuItems/&gt;&lt;/MenuItems&gt;&lt;/Role&gt;&lt;Role
guid=&quot;{8ABB8A10-4418-4467-9E18-99D11FA54E30}&quot;
name=&quot;Manager&quot;&gt;&lt;Apps&gt;&lt;Application productId=&quot;{5B04B775-356B-4AA0-
AAF8-6491FFEA5612}&quot; pinToStart=&quot;1&quot;/&gt;&lt;/Apps&gt;&lt;Settings&gt;&lt;System
name=&quot;Microsoft.Themes&quot;
/&gt;&lt;/Settings&gt;&lt;Buttons&gt;&lt;ButtonLockdownList&gt;&lt;Button
name=&quot;Start&quot;&gt;&lt;ButtonEvent name=&quot;Press&quot; /&gt;&lt;ButtonEvent
name=&quot;PressAndHold&quot;
/&gt;&lt;/Button&gt;&lt;/ButtonLockdownList&gt;&lt;ButtonRemapList/&gt;&lt;/Buttons&gt;&lt;Men
uItems&gt;&lt;DisableMenuItems/&gt;&lt;/MenuItems&gt;&lt;/Role&gt;&lt;/RoleList&gt;&lt;/Handhe
ldLockdown&gt;"/>
    </characteristic>
  </characteristic>
</wap-provisioningdoc>
```

Language

The following example shows how to specify the language to display on the device.

```
<wap-provisioningdoc>
   <characteristic type="EnterpriseAssignedAccess">
  <characteristic type="Language">
      <parm name="Language" datatype="string"
   <parm name="Language" value="1033" />
   </characteristic>
</wap-provisioningdoc>
```

OMA DM examples

These XML examples show how to perform various tasks using OMA DM.

Assigned access settings

The following example shows how to lock down a device.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Add>
         <CmdID>2</CmdID>
         <Item>
            <Target>

<LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/AssignedAccess/AssignedAccessXml</LocURI>
            </Target>
            <Data>&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-
8&quot;?&gt;&lt;HandheldLockdown
version=&quot;1.0&quot;&gt;&lt;Default&gt;&lt;Apps&gt;&lt;Application
productId=&quot;{5B04B775-356B-4AA0-AAF8-6491FFEA5615}&quot;
pinToStart=&quot;1&quot;/&gt;&lt;Application productId=&quot;{5B04B775-356B-4AA0-AAF8-
6491FFEA5612}&quot; pinToStart=&quot;2&quot;/&gt;&lt;/Apps&gt;&lt;Settings&gt;&lt;System
name=&quot;Microsoft.Themes&quot; /&gt;&lt;System name=&quot;Microsoft.About&quot;
/&gt;&lt;/Settings&gt;&lt;Buttons&gt;&lt;Button name=&quot;Start&quot;
disableEvents=&quot;PressAndHold&quot; /&gt;&lt;Button name=&quot;Camera&quot;
disableEvents=&quot;All&quot; /&gt;&lt;Button name=&quot;Search&quot;
disableEvents=&quot;All&quot;
/&gt;&lt;/Buttons&gt;&lt;MenuItems&gt;&lt;DisableMenuItems/&gt;&lt;/MenuItems&gt;&lt;/Default&
gt;&lt;RoleList&gt;&lt;Role guid=&quot;{76C01983-A872-4C4E-B4C6-321EAC709CEA}&quot;
name=&quot;Associate&quot;&gt;&lt;Apps&gt;&lt;Application productId=&quot;{5B04B775-356B-4AA0-
AAF8-6491FFEA5615}&quot; pinToStart=&quot;1&quot;/&gt;&lt;/Apps&gt;&lt;Settings&gt;&lt;System
name=&quot;Microsoft.Themes&quot; /&gt;&lt;System name=&quot;Microsoft.About&quot;
/&gt;&lt;/Settings&gt;&lt;Buttons&gt;&lt;Button name=&quot;Start&quot;
disableEvents=&quot;PressAndHold&quot; /&gt;&lt;Button name=&quot;Camera&quot;
disableEvents=&quot;All&quot;
/&gt;&lt;/Buttons&gt;&lt;MenuItems&gt;&lt;DisableMenuItems/&gt;&lt;/MenuItems&gt;&lt;/Role&gt;
&lt;Role guid=&quot;{8ABB8A10-4418-4467-9E18-99D11FA54E30}&quot;
name=&quot;Manager&quot;&gt;&lt;Apps&gt;&lt;Application productId=&quot;{5B04B775-356B-4AA0-
AAF8-6491FFEA5612}&quot; pinToStart=&quot;1&quot;/&gt;&lt;/Apps&gt;&lt;Settings&gt;&lt;System
name=&quot;Microsoft.Themes&quot; /&gt;&lt;/Settings&gt;&lt;Buttons&gt;&lt;Button
name=&quot;Start&quot; disableEvents=&quot;PressAndHold&quot;
/&gt;&lt;/Buttons&gt;&lt;MenuItems&gt;&lt;DisableMenuItems/&gt;&lt;/MenuItems&gt;&lt;/Role&gt;
&lt;/RoleList&gt;&lt;/HandheldLockdown&gt;</Data>
         </Item>
      </Add>
      <Final/>
   </SyncBody>
</SyncML>
```

Theme

The following example shows how to change the accent color to one of the standard colors.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Replace>
         <CmdID>1</CmdID>
         <Item>
            <Target>
             <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/Theme/ThemeAccentColorID</LocURI>
            </Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <!-- zero based index of available theme colors -->
            <Data>7</Data>
         </Item>
      </Replace>
      <Final/>
   </SyncBody>
</SyncML>
```

The following example shows how to change the theme.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
       <Replace>
           <CmdID>1</CmdID>
           <Item>
               <Target>

<LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/Theme/ThemeBackground</LocURI>
               </Target>
               <Meta>
                   <Format xmlns="syncml:metinf">int</Format>
               </Meta>
               <!-- 0 for "light", 1 for "dark" -->
               <Data>1</Data>
           </Item>
       </Replace>
       <Final/>
   </SyncBody>
</SyncML>
```

The following example shows how to set a custom theme accent color for the enterprise environment.

```
<SyncBody>
   <Replace>
      <CmdID>1</CmdID>
      <Item>
         <Target>
             <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/Theme/ThemeAccentColorID</LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">int</Format>
         </Meta>
         <!—set to Enterprise custom -->
         <Data>151</Data>
      </Item>
   </Replace>
   <Replace>
      <CmdID>2</CmdID>
      <Item>
```

```
         <Target>

<LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/Theme/ThemeAccentColorValue</LocURI>
         </Target>
         <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
         </Meta>
         <!—sets custom accent color of red -->
         <Data>FF0000</Data>
      </Item>
   </Replace>
   <Final/>
</SyncBody>
```

Lock screen

Use the examples in this section to set a new lock screen and manage the lock screen features. If using a UNC path, format the LocURI as \\host\share\image.jpg.

```
<Add>
  <CmdID>2</CmdID>
  <Item>
    <Target>
      <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/LockScreenWallpaper/BGFileName</LocURI>
    <Meta>
      <Format xmlns="syncml:metinf">chr</Format>
      <Type xmlns="syncml:metinf">text/plain</Type>
    </Meta>
    <Data>c:\windows\system32\lockscreen\480x800\Wallpaper_015.jpg </Data>
    </Target>
  </Item>
</Add>
```

The following example shows how to query the device for the file being used as the lock screen.

```
<Get>
  <CmdID>2</CmdID>
  <Item>
    <Target>
      <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/LockScreenWallpaper/BGFileName</LocURI>
    </Target>
  </Item>
</Get>
```

The following example shows how to change the existing lock screen image to one of your choosing.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Replace>
         <CmdID>2</CmdID>
         <Item>
            <Target>

<LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/LockScreenWallpaper/BGFileName</LocURI>
            </Target>
            <Meta>
               <Format xmlns="syncml:metinf">chr</Format>
               <Type xmlns="syncml:metinf">text/plain</Type>
            </Meta>
            <Data>c:\windows\system32\lockscreen\480x800\Wallpaper_015.jpg</Data>
         </Item>
      </Replace>
```

```
      <Final/>
   </SyncBody>
</SyncML>
```

Persist provisioned data

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Add>
         <CmdID>2</CmdID>
         <Item>
           <Target>
            <LocURI>./Vendor/MSFT/EnterpriseEAssignedAccess/PersistData/PersistProvisionedData
</LocURI>
           </Target>
           <Meta>
              <Format xmlns="syncml:metinf">int</Format>
           </Meta>
           <Data>1</Data>
         </Item>
      </Add>
      <Final/>
   </SyncBody>
</SyncML>
```

Time zone

The following example shows how to set the time zone to UTC-07 Mountain Time (US & Canada).

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Replace>
         <CmdID>2</CmdID>
         <Item>
            <Target>
               <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/TimeZone</LocURI>
            </Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>500</Data>
         </Item>
      </Replace>
      <Final/>
   </SyncBody>
</SyncML>
```

The following example shows how to set the time zone to Pacific Standard Time (UTC-08:00) without observing daylight savings time (UTC+01:00).

```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Replace>
         <CmdID>2</CmdID>
         <Item>
            <Target>
               <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/TimeZone</LocURI>
            </Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>400 </Data>
```

```
         </Item>
      </Replace>
      <Final/>
   </SyncBody>
</SyncML>
```

Language

The following example shows how to set the language.

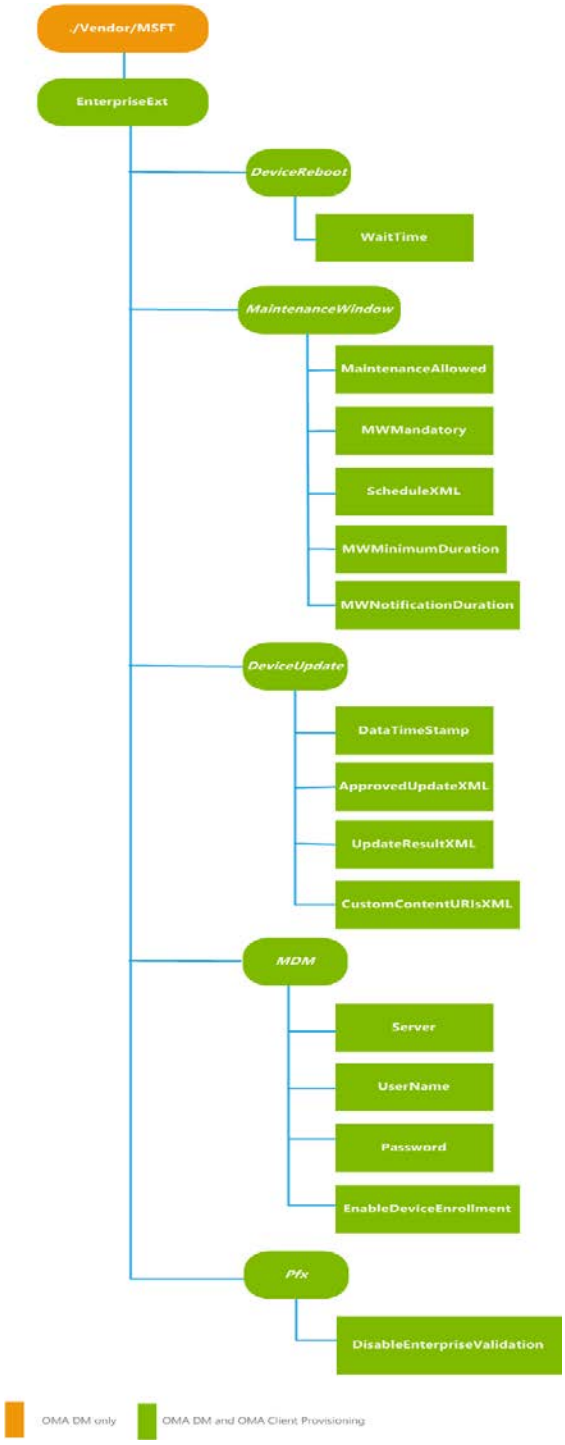
```
<SyncML xmlns="SYNCML:SYNCML1.2">
   <SyncBody>
      <Replace>
         <CmdID>1</CmdID>
         <Item>
            <Target>
               <LocURI>./Vendor/MSFT/EnterpriseAssignedAccess/Locale/Language</LocURI>
            </Target>
            <Meta>
               <Format xmlns="syncml:metinf">int</Format>
            </Meta>
            <Data>1033</Data>
         </Item>
      </Replace>
      <Final/>
   </SyncBody>
</SyncML>
```

EnterpriseExt configuration service provider (Handheld 8.1)

The EnterpriseExt configuration service provider allows Information Technology (IT) administrators to use the Mobile Device Management (MDM) service to set up a device to enroll automatically to the MDM server in an enterprise environment, restart a device, and manage the maintenance window schedule for a device so that it can perform device updates and other management tasks. IT administrators can also test updates with a small number of devices in their environment before deploying the same update to all devices.

Important note: This CSP applies only to Windows Embedded 8.1 Handheld devices.

The following image shows the EnterpriseExt configuration service provider in tree format as used by both the Open Mobile Alliance (OMA) Device Management (DM) and OMA Client Provisioning.


./Vendor/MSFT
EnterpriseExt
DeviceReboot
WaitTime
MaintenanceWindow
MaintenanceAllowed
MWMandatory
ScheduleXML
MWMinimumDuration
MWNotificationDuration
DeviceUpdate
DataTimeStamp
ApprovedUpdateXML
UpdateResultXML
CustomContentURIsXML
MDM
Server
UserName
Password
EnableDeviceEnrollment
Pfx
DisableEnterpriseValidation
OMA DM only
OMA DM and OMA Client Provisioning

./Vendor/MSFT/EnterpriseExt

The root node for the EnterpriseExt configuration service provider. Supported operations: Add, Replace, and Get.

DeviceReboot

The root node for the device reboot command. Supported operations: Exec.

DeviceReboot/WaitTime

The number of seconds, from 0 to 86400, to wait before restarting the device after the exec command is received for DeviceReboot.

The following table shows the possible values and actions:

Value	Action
0	Restart immediately
1 to 300	The device will be restarted silently after the specified amount of time elapses.
300 to 86400	The user will be prompted 5 minutes (300 seconds) prior to restart with an option to restart now: • If the user chooses Cancel, the device will restart after waiting for 5 minutes. • If the user chooses OK, the device restarts immediately.

MaintenanceWindow

The root node for the maintenance window.

The MDM server queries the device to see if it is in a maintenance window so that it can be updated. If the device is not in a maintenance window, the server blocks maintenance actions that might shut down or restart the device. If the device is in a maintenance window, the server does not block the actions.

Before a maintenance window begins on the device, the user is notified that it will soon begin. Because the scheduler manages the notification window and only one instance of the scheduler can run, a command that an IT administrator pushes to the device will be run after the notification window is closed.

To change the duration of a maintenance window while a device is in the maintenance window, the IT administrator must delete the existing maintenance window configuration and then apply the new maintenance window configuration to the device. The new configuration can change the duration of the current maintenance window, but the device will remain in a maintenance window even if the new configuration includes a later start time.

The state of being in a maintenance window cannot be cancelled while it is in progress.

MaintenanceWindow/MaintenanceAllowed

Specifies whether maintenance is allowed on the device: a value of 1 indicates that maintenance is allowed because the device is either in a maintenance window or no maintenance window is scheduled. A value of 0 indicates that no maintenance is allowed because the device is outside of the scheduled maintenance window.

Supported operations: Get.

MaintenanceWindow/MWMandatory

Returns 1 if a maintenance window is mandatory or 0 if a maintenance window is optional (from an end-user perspective). This is a global setting that affects all the scheduled maintenance windows on the device.

Sets the value for Boolean flag that indicates whether a maintenance window is mandatory and whether a user should be able to cancel the maintenance window. The default value is 1, which indicates that the maintenance window is cannot be cancelled by user. This is a global setting that affects all the scheduled maintenance windows on the device.

Supported operations: Get, Set.

MaintenanceWindow/ScheduleXML

Gets the list of maintenance window schedules as XML.

Replaces the current schedule with a new set of schedules or adds new schedules if there are no existing schedules. Replace is destructive and will erase the old schedule.

Supported operations: Add, Replace, and Get.

Use the parameters in the following table to specify the schedule for the maintenance window.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Enabled</td><td>Specify true to enable the maintenance window; otherwise, specify false.</td></tr>
<tr><td>Schedule StartDate</td><td>Specifies the date to start the scheduled maintenance window, including the year, month, day, T as a separator, hour, minute, and second: YYYY-MM-DDTHH:mm:ss format. For example: 2013-10-27T02:00:00</td></tr>
<tr><td>IsUTC</td><td>Specify true if the time should be interpreted as UTC time; otherwise, specify false.</td></tr>
<tr><td>Duration</td><td>Required. Specifies the duration of the scheduled maintenance window:
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Days</td><td>An integer that specifies the number of days: 01 through 31.</td></tr>
<tr><td>Hours</td><td>An integer that specifies the number of hours for the maintenance window: 0 through 23.</td></tr>
<tr><td>Minutes</td><td>An integer that specifies the number of minutes in addition to the hours for the maintenance window: 0 through 59.</td></tr>
</table></td></tr>
<tr><td>Recurrence</td><td>Required. Specifies the recurrence schedule for the maintenance window:
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Type</td><td>Required. A string that specifies the recurrence schedule:
None – one time instance only.
Interval—occurs at a specified interval as defined by MinuteSpan.</td></tr>
<tr><td>MinuteSpan</td><td>Required if the type is Interval. Specifies the interval recurrence in minutes: 0 through 59.</td></tr>
<tr><td>HourSpan</td><td>Required if the type is Interval. Specifies the interval recurrence in hours: 0 through 23.</td></tr>
<tr><td>DaySpan</td><td>Required if the type is Interval. Specifies the interval recurrence in days: 0 through 31.</td></tr>
</table></td></tr>
</table>

MaintenanceWindow/MWNotificationDuration

Gets the duration of pop-up windows in minutes.

Sets the pop-up window duration in minutes. The default duration is 5 minutes.

Supported operations: Get, Set.

MaintenanceWindow/MWminimumDuration

Gets the minimum duration to be considered as a valid MaintenanceWindow.

Sets the minimum duration of maintenance window to be considered a valid window by the device. The default minimum duration is 5 minutes.

Supported operations: Get, Set.

DeviceUpdate

The parent node for device update settings.

DeviceUpdate/DateTimeStamp

Specifies when to start a new device update session. This number can be any integer greater than zero, but to start a new session, the value must be greater than the value that is on the device.

Supported operations: Add, Replace.

DeviceUpdate/UpdateResultXml

Specifies the update information that is on the device.

Supported operations: Add, Get.

MDM

The parent node for MDM settings.

MDM/Server

A string that specifies the MDM server to enroll the device to.

MDM/Username

A string that specifies the username of the person to enroll.

To enroll multiple devices (bulk enroll), enter the same user credentials for all devices.

MDM/Password

A string that specifies the password for the person to enroll.

MDM/EnableDeviceEnrollment

Set to true to skip the discovery service; otherwise, set to false. The default value is true.

This value is saved in the HKLM\Software\Microsoft\Enrollment\EnableDeviceEnrollment registry key, and is set to a pre-defined value of 0 (zero).

Pfx

The parent node for enterprise certificate validation.

DisableEnterpriseValidation

Set to true to disable validation of certificates installed on the device; if set to false or if the value is not set, then the validation of certificates is performed by contacting the Microsoft server over the Internet. If the device does not have an Internet connection and the value is set to false, then apps may be disabled or deployments may be blocked.

Value type is Boolean.

This value is saved in the registry in the HKLM\software\microsoft\enterpriseappmanagement\appmanagementvalidation\config\DisabledByEnterprise key.

The restart process

1. The restart command is sent as an XML provisioning file to the device.
2. The user is alerted that the company IT requires that the device be restarted, and the device will be restarted after waiting for the number of seconds specified in DeviceReboot/WaitTime.

The enrollment process

1. When you load the provisioning file to a device during the OOBE by using a Secure Digital (SD) card or near-field communication (NFC) Tag, the device connects to the company Wi-Fi that is defined in the provisioning file and then initiates the enrollment to the MDM server.
2. The device sends unique and constant device IDs to the MDM server as part of the enrollment information.
3. The server then authenticates the device

OMA client provisioning examples

The XML examples in this section show how to perform various tasks by using OMA client provisioning.

MDM enrollment example

The following example shows how to enroll a device.

```
<characteristic type="EnterpriseExt">
<characteristic type="MDM">
   <parm value="contoso.com:443" name="Server"/>
   <parm value="@contoso.com" name="Username"/>username<parm value="password"
name="Password"/>
   <parm value="TRUE" name="EnableDeviceEnrollment"/>
</characteristic>
```

OMA DM examples

These XML examples show how to perform various tasks using OMA DM.

Device restart example

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Exec>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/DeviceReboot/WaitTime</LocURI>
        </Target>
        <Meta>
          <Format xmlns="syncml:metinf">int</Format>
        </Meta>
        <Data>0</Data>
      </Item>
    </Exec>
  </SyncBody>
</SyncMl>
```

Maintenance window examples

Set the maintenance window schedule. In this example, the schedule starts on October 27, 2013 (10/27/2013) at 2:00 A.M. and lasts for 4 hours:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Add>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/ScheduleXml</LocURI>
        </Target>
        <Meta>
          <Format xmlns="syncml:metinf">chr</Format>
        </Meta>
        <Data>&lt;MWList&gt;&lt;MW Enabled=&quot;True&quot;&gt;&lt;Schedule
StartDate=&quot;2013-10-27T02:00:00&quot; IsUTC=&quot;False&quot;/&gt;&lt;Duration
Days=&quot;0&quot; Hours=&quot;4&quot; Minutes=&quot;0&quot;/&gt;&lt;Recurrence
Type=&quot;None&quot; MinuteSpan=&quot;0&quot; HourSpan=&quot;0&quot;
DaySpan=&quot;0&quot;/&gt;&lt;/Schedule&gt;&lt;/MW&gt;&lt;/MWList&gt;
        </Data>
      </Item>
    </Add>
  </SyncBody>
</SyncMl>
```

Retrieve the maintenance window schedule:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Get>
      <CmdID>3</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/ScheduleXml</LocURI>
        </Target>
      </Item>
    </Get>
  </SyncBody>
</SyncMl>
```

Retrieve the **MaintenanceAllowed** value:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Get>
      <CmdID>3</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MaintenanceAllowed</LocURI>
        </Target>
      </Item>
    </Get>
  </SyncBody>
</SyncMl>
```

Set the **MWNotificationDuration** value to 3 minutes:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Replace>
```

```
      <CmdID>2</CmdID>
      <Item>
        <Target>

<LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MWNotificationDuration</LocURI>
        </Target>
        <Meta>
          <Format xmlns="syncml:metinf">int</Format>
        </Meta>
        <Data>3</Data>
      </Item>
    </Replace>
</SyncBody>
</SyncMl
```

Set the **MWMinimumDuration** to 2 hours:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Replace>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MWMinimumDuration</LocURI>
        </Target>
        <Meta>
          <Format xmlns="syncml:metinf">int</Format>
        </Meta>
        <Data>120</Data>
      </Item>
    </Replace>
</SyncBody>
</SyncMl
```

Set **MWMandatory** to 0, which means that the window can be cancelled by the device user:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Replace>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MWMandatory</LocURI>
        </Target>
        <Meta>
          <Format xmlns="syncml:metinf">int</Format>
        </Meta>
        <Data>0</Data>
      </Item>
    </Replace>
</SyncBody>
</SyncMl
```

Get the current values of MWNotificationDuration, MWMinimumDuration, and MWMandatory values from the device:

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Get>
      <CmdID>1</CmdID>
      <Item>
        <Target>
```

```
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MWNotificationDuration
</LocURI>
        </Target>
      </Item>
    </Get>
    <Get>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MWMinimumDuration</LocURI>
        </Target>
      </Item>
    </Get>
    <Get>
      <CmdID>3</CmdID>
      <Item>
        <Target>
          <LocURI>./Vendor/MSFT/EnterpriseExt/MaintenanceWindow/MWMandatory</LocURI>
        </Target>
      </Item>
    </Get>
  </SyncBody>
</SyncML>
```

Schema for the maintenance ScheduleXML parameters

```
<?xml version="1.0" encoding="utf-16LE" ?>
<!--
     In-memory format is Little Endian and
     hence the encoding of this file has to be little endian
     to be in the native format. Make sure that this file's
     encoding is Unicode-16 LE (Unicode Codepage 1200)  --> <xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
>
  <!-- COMPLEX TYPE: Duration -->
  <xs:complexType name="duration_t">
    <xs:attribute name="Days" >
      <xs:simpleType>
        <xs:restriction base="xs:unsignedInt">
          <xs:minInclusive value="0" />
          <xs:maxInclusive value="31" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Hours" >
      <xs:simpleType>
        <xs:restriction base="xs:unsignedInt">
          <xs:minInclusive value="0" />
          <xs:maxInclusive value="23" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Minutes" >
      <xs:simpleType>
        <xs:restriction base="xs:unsignedInt">
          <xs:minInclusive value="0" />
          <xs:maxInclusive value="59" />
        </xs:restriction>
      </xs:simpleType>
```

```
  </xs:attribute>
</xs:complexType>

<!-- COMPLEX TYPE: Recurrence -->
<xs:complexType name="recurrence_t">
  <xs:attribute name="Type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="None|Interval"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="MinuteSpan" >
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="59" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="HourSpan" >
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="23" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="DaySpan" >
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="31" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<!-- COMPLEX TYPE: Schedule TYPE -->
<xs:complexType name="schedule_t">
  <xs:sequence>
    <xs:element name="Duration" type="duration_t" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Recurrence" type="recurrence_t" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="StartDate" type="xs:dateTime" use="required"/>
  <xs:attribute name="IsUTC" type="xs:boolean" use="required"/>
</xs:complexType>

<!-- COMPLEX TYPE: PROP TYPE -->
<xs:complexType name="prop_t">
  <xs:attribute name="Name" type="xs:string"/>
  <xs:attribute name="Value" type="xs:string"/>
  <xs:attribute name="Datatype" >
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="integer|string|boolean"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
```

```
  </xs:complexType>

  <!-- COMPLEX TYPE: PropList TYPE -->
  <xs:complexType name="proplist_t">
    <xs:sequence>
      <xs:element name="Prop" type="prop_t" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!-- COMPLEX TYPE: MW TYPE -->
  <xs:complexType name="mw_t">
    <xs:sequence>
      <xs:element name="Schedule" type="schedule_t" minOccurs="1" />
    </xs:sequence>
    <xs:attribute name="Enabled" type="xs:boolean" use="required"/>
  </xs:complexType>

  <!-- SCHEMA -->
  <xs:element name="MWList">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PropList" type="proplist_t" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="MW" type="mw_t" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" use="required" type="xs:decimal"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

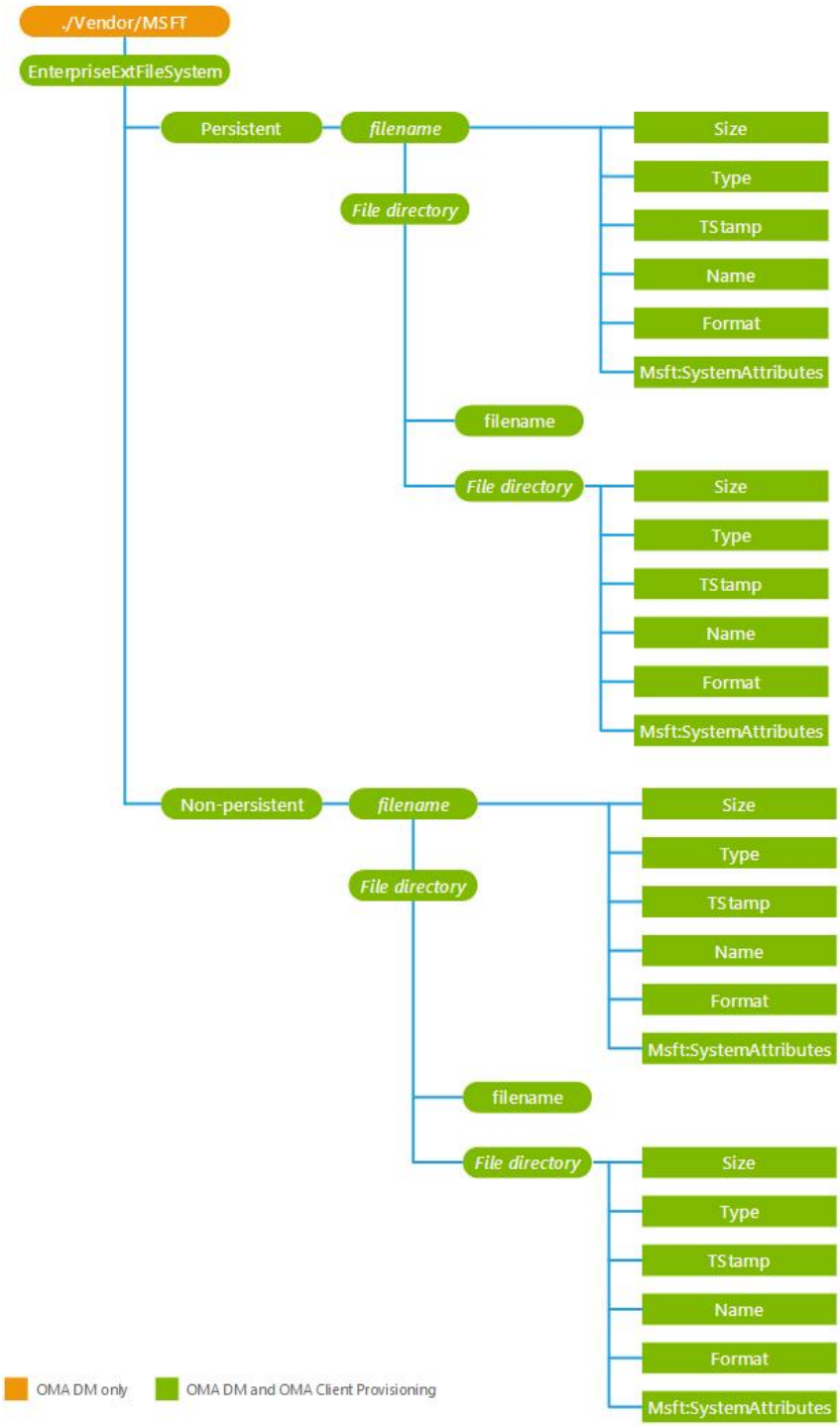
EnterpriseExtFileSystem configuration service provider (Handheld 8.1)

The EnterpriseExtFileSystem configuration service provider (CSP) allows Information Technology (IT) administrators to add, retrieve, or change files in the file system through the Mobile Device Management (MDM) service. For example, you can use this configuration service provider to push a provisioning XML file or a new lock screen background image file to a device through the MDM service, and also retrieve logs from the device in the enterprise environment.

Important note: This CSP applies only to Windows Embedded 8.1 Handheld devices.

File contents are embedded directly into the syncML message, so there is a limit to the size of the file that can be retrieved from the device. The default limit is 100000 (1 MB). You can configure this limit by using the following registry key:
Software\Microsoft\Provisioning\CSPs\.\Vendor\MSFT\EnterpriseExtFileSystem\MaxFileReadSize.

The following image shows the EnterpriseExtFileSystem configuration service provider in tree format as used by the Open Mobile Alliance (OMA) Device Management (DM).


./Vendor/MSFT
EnterpriseExtFileSystem
Persistent
filename
Size
Type
TStamp
Name
Format
Msft:SystemAttributes
File directory
filename
File directory
Size
Type
TStamp
Name
Format
Msft:SystemAttributes
Non-persistent
filename
Size
Type
TStamp
Name
Format
Msft:SystemAttributes
File directory
filename
File directory
Size
Type
TStamp
Name
Format
Msft:SystemAttributes
OMA DM only
OMA DM and OMA Client Provisioning

./Vendor/MSFT/EnterpriseExtFileSystem

The root node for the EnterpriseExtFileSystem configuration service provider. Supported operations: add and get.

Persistent

The EnterpirseExtFileSystem CSP allows an enterprise to read, write, delete and list files in this folder.

Anything stored in the persistent folder can be backed up before a device is wiped. If it is backed up, it will be restored when the device boots again.

NonPersistent

The EnterpirseExtFileSystem CSP allows an enterprise to read, write, delete and list files in this folder.

Anything stored in the NonPersistent folder will be deleted the next time the device is wiped.

<File directory>

The name of a directory in the device file system. Any <file directory> node can have directories and files as child nodes.

Use the add command to create a new directory. You cannot use it to add a new directory under a file system root.

Use the get command to return the list of child node names under <file directory>.

Use the get command with ?List=Struct to recursively return all child node names, including subdirectory names, under <file directory>.

<file name>

The name of a file in the device file system.

Supported operations: get.

The following table shows supported characteristics for files and directories.

Property	Description
Name	Supported operations: **get**. The **get** command returns the name of the file or file directory.
Format	For a directory, specify node. For a file, leave blank. Supported operations: **get**. For files, when binary data is sent over XML, it is Base64 encoded. When binary data is sent over wbxml, bin format is used directly.
Type	Supported operations: **get**. For the FileSystem root node, the **get** command returns for the object identifier similar to the following: com.microsoft/windowsmobile/1.1/FileSystemMO The **get** command returns blank for all other file directory nodes. For files, the **get** command specifies application/octet-stream as the MIME

	type of the file. The configuration service provider treats all files as a binary data block.
TStamp	Supported operations: **get**. The **get** command returns data about the last time the directory or file was changed. The value is represented by a string that contains a UTC-based, ISO 8601 basic format, complete representation of a date and time value. For example, 20120711T163817Z means July 11, 2012 at 16 hours, 38 minutes, and 17 seconds.
Size	Supported operations: **get**. This parameter is not supported in a file directory. For files, the **get** command returns the file content size in bytes. For a binary file, the size is for the unencoded file.
msft:SystemAttributes	Supported operations: **get** and **replace**. A custom property created by Microsoft that contains directory attributes. The **get** command returns the file or file directory attributes. The **replace** command changes the file attributes.
msft:AccessRoles	Not supported.

OMA DM examples

The following example shows how to retrieve a file from the device.

```
<Get>
    <CmdID>2</CmdID>
    <Item>
        <Target>
            <LocURI>./Vendor/MSFT/EnterpriseExtFileSystem/C%3A/data/test/bin/file.txt</LocURI>
        </Target>
    </Item>
</Get>
```

The following example shows the file name that is returned in the body of the response syncML code. In this example, the full path of the file on the device is C:/data/test/bin/filename.txt.

```
<Results>
    <CmdID>3</CmdID>
    <MsgRef>1</MsgRef>
    <CmdRef>2</CmdRef>
    <Item>
        <Source>

<LocURI>./Vendor/MSFT/EnterpriseExtFileSystem/C%3A/data/test/bin/filename.txt</LocURI>
        </Source>
        <Meta>
            <Format xmlns="syncml:metinf">b64</Format>
            <Type xmlns="syncml:metinf">application/octet-stream</Type>
        </Meta>
        <Data>aGVsbG8gd29ybGQ=</Data>
```

```
    </Item>
</Results>
```

The following example shows how to push a file to the device.

```
<Add>
   <CmdID>2</CmdID>
   <Item>
      <Target>
         <LocURI>./Vendor/MSFT/EnterpriseExtFileSystem/C%3A/data/test/bin/new.txt</LocURI>
      </Target>
      <Meta>
          <Format xmlns="syncml:metinf">b64</Format>
          <Type xmlns="syncml:metinf">application/octet-stream</Type>
      </Meta>
      <Data>aGVsbG8gd29ybGQ=</Data>
   </Item>
</Add>
```

Reference

- [MS-XCEP]: X.509 Certificate Enrollment Policy Protocol Specification
- [MS-WSTEP]: WS-Trust X.509v3 Token Enrollment Extensions
- OMA Device Management Protocol v1.2
- OMA Device Management Security
- OMA DM Standardized Objects
- OMA DM Representation protocol
- OMA DM Tree and Description
- OMA DM Bootstrap
- Application Characteristic for OMA Device Management
- [SCEP]: Simple Certificate Enrollment Protocol

Q&A

This section lists the common questions MDM vendors may have and corresponding answers.

Question	Answer
Can enrollment be initiated via SMS?	Not supported in Windows Phone 8 and 8.1.
Can the DM client support using a proxy to make connections to the MDM server for authentication and check-in?	HTTP proxy over Wi-Fi is supported. Proxy authentication isn't supported in Windows Phone 8 and 8.1.
Do you support SSL offloading?	Not supported in Windows Phone 8 and 8.1.

Question	Answer
Will the discovery request accept self-signed certs or prompt the user?	While the discovery request doesn't reject self-signed certs, and the phone prompts the user for permission to continue, if server use WAB to get security token, WAB server certificate must root to a device known root certificate The user could manually install root certificate before MDM enrollment
Can a phone to be enrolled with multiple companies? Can a user create multiple "company apps" accounts?	In Windows Phone 8 and 8.1, we support one company account for the enterprise device management server. If desired, the user can acquire and install multiple companies' enterprise application tokens. They can then manually install apps that depend on those tokens alongside apps installed via the enrolled company account.
Can a phone be owned by multiple users?	One phone could be used by multiple users. However, at any one time, there is only one company apps account. The user needs to delete the old account used by another user and add a new account.
Unenrollment/disassociation seems to happen during a normal maintenance session - can an unenrollment command be pushed to the phone?	Server push support is not supported in Windows Phone 8. Server push via WNS is supported in Windows Phone 8.1.
Are the enterprise apps installed through the MDM server removed when the phone is unenrolled?	Yes.
Can a user be prevented from unenrolling or unregistering?	Not supported in Windows Phone 8. The user has the final authority to decide whether to disassociate the phone from company use. For Windows Phone 8.1, the MDM server could push down a policy to disallow the user to unenroll the created workplace account from UI.
What is the behavior when the phone is locked? Is it going to connect to the MDM server when it is locked at a sync interval? If so, can we push enterprise profiles when the phone is locked?	The device lock will not prevent the phone from connecting back to the MDM server at the scheduled sync interval. So, yes, you can push settings to the phone when the phone is locked.
What kinds of errors are reported by the phone while the MDM server configures the phone?	During a DM session, the phone reports various error codes depending on which DM command is sent by the server. For information about common errors the phone could send, see the OMA DM representation doc. Most are very straightforward.
Is there any way we can send user-facing messages to the user?	Not supported in Windows Phone 8 and 8.1.
Can the MDM server get the location of the phone from the MDM client?	Not supported in Windows Phone 8.1.

Question	Answer
When a certificate expires, the client can be offline or the phone can be turned off. Does letting the certificate expire require the user to wipe the MDM relationship (along with all the apps and settings)?	When the certificate expires, the relationship is not wiped. But the user cannot launch the installed enterprise app and the client cannot communicate with the server (client authentication will fail) until the certificate is renewed if manual renew is supported by the server or the user needs to unenroll and re-enroll to be MDM managed again if automatic renew is supported by the server.
Is it possible to get an application inventory of the entire phone or just enterprise-deployed apps?	The MDM server can only inventory enterprise-deployed apps. However if the server knows the product ID of the app, it could query to find out whether that app is installed in the device in Windows Phone 8.1.
Is it possible to get logs for enrollment or an MDM session through either tethering or over the air?	In Windows Phone 8.1, for developer unlocked phone, there is a tool for ISV to get some enrollment and MDM session related log.
Can the DM interval/synch policy/certificate renewal period be changed after enrollment?	These are only set during enrollment in Windows Phone 8.0 In Windows Phone 8.1, MDM client renew retry interval and regular DM client polling schedule is configurable via DM session
Is there an alert outside "company apps" settings to warn the user about required renewal?	Yes, two types of alerts are provided to the user. Before the certificate expires, the phone will prompt the user to go to the settings page to provide an updated password when the company apps account is about to expire. If the user tries to launch an app installed via the Company Hub after the cert is expired, the app cannot be launched and the user will get a notification to go to the settings page to update the company apps account.
What happens to the settings or apps installed through the DM client after the certificate is expired?	Apps will remain on the phone but fail to launch. Policies specified by the MDM server will remain as valid. Reinstalling a valid cert will re-enable the apps.
Other than one mandatory app pushed at the end of enrollment, is there a way to push certain applications on the phone? Automatically push certain apps from the Company Hub?	In Windows Phone 8.0, only a single app can be pushed, which is at the end of enrollment. The MDM server can push updates for LOB apps that have been installed by the user. In Windows Phone 8.1, MDM server could install, update, delete, and query enterprise applications that is signed with company's AET token.

Question	Answer
Is there a unique identifier that could be retrieved by both company applications and the MDM server?	Yes. During enrollment, the server sends down an AET (application enrollment token) to the phone for application distribution from that enterprise. The AET contains the Enterprise ID (also known as Publisher ID). This publisher ID is used to form a publisher-specific phone ID. The MDM server can retrieve the publisher-specific phone ID by querying the following property URI: ./vendor/MSFT/DMClient/Provider/<provider-id>/PublisherDeviceID. Applications can use publisher-specific API to retrieve the same value that the MDM server retrieved right after enrollment - DeviceExtendedProperties ("DeviceUniqueId").
To support push notification for Windows Phone, do MDM ISVs need to get separate credential for Windows and Windows push notification?	No. MDM ISVs get one set of credentials from Microsoft for MDM Push for Windows and Windows Phone, by going and getting one from Windows.
How to differentiate MDM enrollment request between Windows Phone 8.1 and Windows Phone 8?	The <RequestVersion> tag in MDM discovery request message contains version information. For Windows Phone 8.1, the value is 2.0. For Windows Phone 8, the value is 1.0
How could MDM management server know whether the device is Windows Phone 8 or Windows Phone 8.1	The MDM management server could query DevDetail/SwV node to find the device OS version and use that information to identify whether it is Windows Phone 8 or Windows Phone 8.1 device.

Appendix

XSD for ApplicationRestriction policy in PolicyManager

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="AppPolicy_xsd"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://schemas.microsoft.com/phone/2013/policy"
           xmlns="http://schemas.microsoft.com/phone/2013/policy"
           xmlns:m="http://schemas.microsoft.com/phone/2013/policy"
           >

  <!-- Non-empty string must have a non-whitespace character at the beginning and end -->
  <xs:simpleType name="ST_NonEmptyString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="32767"/>
      <xs:pattern value="[^\s]|([^\s].*[^\s])"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_Publisher">
    <xs:restriction base="xs:string">
```

```
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="CT_LowerCaseGuid">
    <xs:annotation>
      <xs:documentation>GUID must use lowercase letters</xs:documentation>
    </xs:annotation>
    <xs:restriction base="ST_NonEmptyString">
      <xs:pattern value="\{[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}\}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="CT_Application">
    <xs:attribute name="ProductId" type="CT_LowerCaseGuid" />
  </xs:complexType>

  <xs:complexType name="CT_ApplicationWithPublisher">
    <xs:attribute name="ProductId" type="CT_LowerCaseGuid" />
    <xs:attribute name="PublisherName" type="ST_Publisher" use="optional" />
  </xs:complexType>

  <xs:complexType name="CT_AllowedPublisher">
    <xs:sequence>
      <xs:element name="DenyApp" type="CT_Application" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="PublisherName" type="ST_Publisher" use="required" />
  </xs:complexType>

  <xs:complexType name="CT_DeniedPublisher">
    <xs:sequence>
      <xs:element name="AllowApp" type="CT_Application" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="PublisherName" type="ST_Publisher" use="required" />
  </xs:complexType>

  <xs:element name="Deny">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="App" type="CT_Application" minOccurs="0" maxOccurs="unbounded" />
        <xs:element name="Publisher" type="CT_DeniedPublisher" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Allow">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="App" type="CT_ApplicationWithPublisher" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element name="Publisher" type="CT_AllowedPublisher" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="AppPolicy">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="1">
        <xs:element ref="Deny" />
        <xs:element ref="Allow" />
```

```
      </xs:choice>
      <xs:attribute name="Version" use="required" type="xs:unsignedLong" />
    </xs:complexType>

    <!-- Uniqueness Checks -->
    <xs:unique name="NoDuplicateProductIDs">
      <xs:selector xpath=".//*"/>
      <xs:field xpath="@ProductId"/>
    </xs:unique>

    <!-- Uniqueness Checks -->
    <xs:unique name="NoDuplicatePublisherNames">
      <xs:selector xpath=".//*"/>
      <xs:field xpath="@PublisherName"/>
    </xs:unique>
  </xs:element>

</xs:schema>
```

XML samples for ApplicationRestriction policy in PolicyManager

Allow List: One allowed application

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: One allowed application and publisher

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
```

```
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - MixRadio --&#x3E;&#x3C;App ProductId=&#x22;{f5874252-1f04-4c3f-a335-
4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Allow Publisher - Microsoft Corporation --
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: One allowed application and two allowed publishers

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Allow Publisher - Microsoft Corporation --
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft Corporation&#x22;/&#x3E;&#x3C;!-- Allow
Publisher - Microsoft Studios&#xE2;&#x201E;&#xA2; --&#x3E;&#x3C;Publisher
PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: One allowed publisher

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
```

```
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow Publisher - Microsoft Corporation --&#x3E;&#x3C;Publisher
PublisherName=&#x22;Microsoft
Corporation&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: Two allowed applications

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Allow App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-
4fa3b7b85329}&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: Two allowed applications and one allowed publisher

```
<SyncML xmlns="SYNCML:SYNCML1.2">
```

```
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Allow App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Allow Publisher -
Microsoft Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: Two allowed applications and two allowed publishers

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Allow App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Allow Publisher -
Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;/&#x3E;&#x3C;!-- Allow Publisher - Microsoft Studios&#xE2;&#x201E;&#xA2;--
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: Two allowed applications, one allowed publisher with one denied application exception, and one allowed publisher with two denied applications exceptions

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Allow App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Allow Publisher -
Microsoft Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;&#x3E;&#x3C;!-- Deny app published by allowed publisher Microsoft Corporation
- Facebook --&#x3E;&#x3C;DenyApp ProductId=&#x22;{82a23635-5bd9-df11-a844-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;!-- Allow Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Deny app published by allowed publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;DenyApp
ProductId=&#x22;{c62201b4-e059-e011-854c-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Deny app
published by allowed publisher Microsoft Studios&#xE2;&#x201E;&#xA2; - Halo: SA Lite --
&#x3E;&#x3C;DenyApp ProductId=&#x22;{cf3f117d-d5a6-4e81-9786-
56dd337b9b02}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data
>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Allow List: Two allowed publishers

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
```

```
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Allow&#x3E;&#x3C;!
-- Allow Publisher - Microsoft Corporation --&#x3E;&#x3C;Publisher
PublisherName=&#x22;Microsoft Corporation&#x22;/&#x3E;&#x3C;!-- Allow Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;/&#x3E;&#x3C;/Allow&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Deny List: Bing News and Skype

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - Bing News --&#x3E;&#x3C;App ProductId=&#x22;{9c3e8cad-6702-4842-8f61-
b8b33cc9caf1}&#x22;/&#x3E;&#x3C;!-- Deny App - Skype --&#x3E;&#x3C;App
ProductId=&#x22;{c3f8e570-68b3-4d6a-bdbb-
c0a3f4360a51}&#x22;/&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Deny List: One denied application and one denied publisher with two allowed application exceptions

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
```

```
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Deny Publisher - Microsoft Corporation --
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft Corporation&#x22;&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Corporation - Facebook --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{82a23635-5bd9-df11-a844-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Corporation - YouTube --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{dcbb1ac6-a89a-df11-a490-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Deny List: One denied application, one denied publisher with one allowed application exception, and one denied publisher with two allowed application exceptions

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - MixRadio --&#x3E;&#x3C;App ProductId=&#x22;{f5874252-1f04-4c3f-a335-
4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Deny Publisher - Microsoft Corporation --
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft Corporation&#x22;&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Corporation - Facebook --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{82a23635-5bd9-df11-a844-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;!-- Deny Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Studios&#xE2;&#x201E;&#xA2; - Halo: SA Lite --
&#x3E;&#x3C;AllowApp ProductId=&#x22;{cf3f117d-d5a6-4e81-9786-
56dd337b9b02}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
```

```
</SyncML>
```

Deny List: One denied application, one denied publishers with two allowed application exceptions, and one denied publishers with two allowed application exceptions

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Deny Publisher - Microsoft Corporation --
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft Corporation&#x22;&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Corporation - Facebook --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{82a23635-5bd9-df11-a844-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Corporation - YouTube --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{dcbb1ac6-a89a-df11-a490-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;!-- Deny Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Studios&#xE2;&#x201E;&#xA2; - Halo: SA Lite --
&#x3E;&#x3C;AllowApp ProductId=&#x22;{cf3f117d-d5a6-4e81-9786-
56dd337b9b02}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Deny List: Two denied applications, one denied publisher, and one denied publisher with one allowed application exception

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Atomic>
            <CmdID>1</CmdID>
            <Replace>
                <CmdID>2</CmdID>
                <Item>
                    <Target>
```

```
<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                    </Target>
                    <Meta>
                        <Format xmlns="syncml:metinf">chr</Format>
                        <Type xmlns="syncml:metinf">text/plain</Type>
                    </Meta>
                    <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Deny App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Deny Publisher -
Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;/&#x3E;&#x3C;!-- Deny Publisher - Microsoft Studios&#xE2;&#x201E;&#xA2;--
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
                </Item>
            </Replace>
        </Atomic>
        <Final/>
    </SyncBody>
</SyncML>
```

Deny List: Two denied applications, one denied publisher with one allowed application exception, and one denied publisher with one allowed application exception

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Atomic>
      <CmdID>1</CmdID>
      <Replace>
        <CmdID>2</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
            <Type xmlns="syncml:metinf">text/plain</Type>
          </Meta>
          <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Deny App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Deny Publisher -
Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher Microsoft Corporation
- YouTube --&#x3E;&#x3C;AllowApp ProductId=&#x22;{dcbb1ac6-a89a-df11-a490-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;!-- Deny Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
        </Item>
```

```
      </Replace>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

Deny List: Two denied applications, one denied publisher with two allowed application exception, and one denied publisher with two allowed application exception

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Atomic>
      <CmdID>1</CmdID>
      <Replace>
        <CmdID>2</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
            <Type xmlns="syncml:metinf">text/plain</Type>
          </Meta>
          <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny App - Nokia Trailers --&#x3E;&#x3C;App ProductId=&#x22;{b0731ce2-cdee-4cad-af01-
a74a0433fcea}&#x22;/&#x3E;&#x3C;!-- Deny App - MixRadio --&#x3E;&#x3C;App
ProductId=&#x22;{f5874252-1f04-4c3f-a335-4fa3b7b85329}&#x22;/&#x3E;&#x3C;!-- Deny Publisher -
Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher Microsoft Corporation
- Facebook --&#x3E;&#x3C;AllowApp ProductId=&#x22;{82a23635-5bd9-df11-a844-
00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app published by denied publisher Microsoft
Corporation - YouTube --&#x3E;&#x3C;AllowApp ProductId=&#x22;{dcbb1ac6-a89a-df11-a490-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;!-- Deny Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Studios&#xE2;&#x201E;&#xA2; - Halo: SA Lite --
&#x3E;&#x3C;AllowApp ProductId=&#x22;{cf3f117d-d5a6-4e81-9786-
56dd337b9b02}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
        </Item>
      </Replace>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

Deny List: one denied publisher and one denied publisher with one allowed application exception

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Atomic>
      <CmdID>1</CmdID>
```

```
      <Replace>
        <CmdID>2</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
            <Type xmlns="syncml:metinf">text/plain</Type>
          </Meta>
          <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny Publisher - Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;/&#x3E;&#x3C;!-- Deny Publisher - Microsoft Studios&#xE2;&#x201E;&#xA2;--
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
        </Item>
      </Replace>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

Deny List: one denied publisher with one allowed application exception and one denid publisher with two allowed application exceptions

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Atomic>
      <CmdID>1</CmdID>
      <Replace>
        <CmdID>2</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
            <Type xmlns="syncml:metinf">text/plain</Type>
          </Meta>
          <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny Publisher - Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher Microsoft Corporation
- Facebook --&#x3E;&#x3C;AllowApp ProductId=&#x22;{82a23635-5bd9-df11-a844-
00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app published by denied publisher Microsoft
Corporation - YouTube --&#x3E;&#x3C;AllowApp ProductId=&#x22;{dcbb1ac6-a89a-df11-a490-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;!-- Deny Publisher - Microsoft
Studios&#xE2;&#x201E;&#xA2;--&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-
00237de2db9e}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
        </Item>
```

```
      </Replace>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

Deny List: one denied publisher with two allowed application exceptions and one denied publisher

```
<SyncML xmlns="SYNCML:SYNCML1.2">
  <SyncBody>
    <Atomic>
      <CmdID>1</CmdID>
      <Replace>
        <CmdID>2</CmdID>
        <Item>
          <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
          </Target>
          <Meta>
            <Format xmlns="syncml:metinf">chr</Format>
            <Type xmlns="syncml:metinf">text/plain</Type>
          </Meta>
          <Data>&#x3C;AppPolicy Version=&#x22;1&#x22;
xmlns=&#x22;http://schemas.microsoft.com/phone/2013/policy&#x22;&#x3E;&#x3C;Deny&#x3E;&#x3C;!-
- Deny Publisher - Microsoft Corporation --&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Corporation&#x22;/&#x3E;&#x3C;!-- Deny Publisher - Microsoft Studios&#xE2;&#x201E;&#xA2;--
&#x3E;&#x3C;Publisher PublisherName=&#x22;Microsoft
Studios&#xE2;&#x201E;&#xA2;&#x22;&#x3E;&#x3C;!-- Allow app published by denied publisher
Microsoft Studios&#xE2;&#x201E;&#xA2; - Wordament --&#x3E;&#x3C;AllowApp
ProductId=&#x22;{c62201b4-e059-e011-854c-00237de2db9e}&#x22;/&#x3E;&#x3C;!-- Allow app
published by denied publisher Microsoft Studios&#xE2;&#x201E;&#xA2; - Halo: SA Lite --
&#x3E;&#x3C;AllowApp ProductId=&#x22;{cf3f117d-d5a6-4e81-9786-
56dd337b9b02}&#x22;/&#x3E;&#x3C;/Publisher&#x3E;&#x3C;/Deny&#x3E;&#x3C;/AppPolicy&#x3E;</Data>
        </Item>
      </Replace>
    </Atomic>
    <Final/>
  </SyncBody>
</SyncML>
```

Known Issues

Support

- If you still have questions after reading through this whitepaper, please use the support forum below created exclusively for MDM development discussions, to post your questions – Developing MDM Solutions
- For 1:1 paid development support, please submit a support incident and choose "Developing MDM Solutions" here: Submit a support ticket.