



BlackBerry AtHoc

SDK Specification

7.7

Contents

- Using the BlackBerry AtHoc SDK APIs..... 5**
 - Common Uses..... 5
 - Using HTTP POST..... 6
 - SDK envelope request..... 7
 - SDK envelope response..... 8
 - SDK payload types..... 10
- Content APIs..... 14**
 - Get distribution lists..... 14
 - Get alert templates..... 14
 - Get alert template..... 16
 - Get hierarchy..... 19
 - Get devices..... 20
 - Get alert folders..... 21
 - Get organization parameters..... 22
- Publish alert API..... 23**
 - Alert payload definition..... 23
 - Publish enterprise alerts..... 43
- Alert tracking API..... 45**
 - Alert list..... 45
 - Summary report..... 47
 - Detailed report by users and devices..... 49
- Distribution lists synchronization..... 51**
 - Distribution lists definition XML..... 51
 - Distribution lists synchronization response..... 53
- User synchronization..... 55**
 - Overview..... 55
 - User synchronization process..... 56
 - User synchronization..... 57
 - Get user details..... 60
 - User device subscription..... 61
- Diagnostic log..... 63**
 - Log diagnostic information..... 63

Manage API Access and Security.....	65
Validate SDK requests.....	65
Validate with a password and username.....	65
Validate with a session ID.....	65
Validate with an access token.....	66
Restrict API access by IP.....	66
Sample code.....	68
C#.....	68
VBScript.....	69
Internationalization-supported characters.....	71
Phone number validation.....	73
BlackBerry AtHoc customer portal.....	75
Legal notices.....	76

Using the BlackBerry AtHoc SDK APIs

BlackBerry AtHoc provides the industry-leading enterprise-class notifications management and delivery software. Incorporating multiple notification sources and a variety of delivery channels, BlackBerry AtHoc products empower organizations to notify targeted audiences of critical, time-sensitive information and workflow events. As a result, enterprises enable target audiences to act on information and events at the moment of high importance, driving business productivity and revenues.

BlackBerry AtHoc provides a suite of portable HTTP and XML-based APIs to facilitate integration of the BlackBerry AtHoc system with external systems, and provides content synchronization, group and distribution list synchronization, and publishing of alerts.

This document describes the suite of API calls, and provides both a technical discussion of how the APIs work and how to use them. It is assumed that the reader is familiar with the BlackBerry AtHoc product, the end-user interaction, and the use of the Management System.

The AtHoc SDK APIs are based on XML over HTTP. The two main methods to use the API are:

- By HTTP POSTing the XML document to AtHoc SDK listener. This method allows sending several API requests (payloads) in one XML document by wrapping the different requests in an SDK envelope. The SDK envelope contains authentication and identification parameters for the SDK request(s).
- In some cases, BlackBerry AtHoc can be configured to pull the data from a configured URL. The pulling context determines the expected data format, and only the payload needs to be transferred from the data source.

The following sections describe how to connect to the AtHoc SDK and the SDK envelope used for posting SDK requests.

Common Uses

Quick publish

To start using publishing APIs for simple use cases, use the following table as a quick reference.

Process	API call	Notes
Show a list of alert templates defined within BlackBerry AtHoc.	GET-SCENARIOS	You can fetch only quick publish alert template or all alert templates.
Choose an alert template to publish; show alert template details.	GET-SCENARIO	—
Publish a given alert template as-is or with edits.	INFOCASTING	type="ALERT ALERTSCENARIO" action="PUBLISH"
Put a draft alert into standby mode so that it can be reviewed or held until a later time.	INFOCASTING	type="ALERT" action="STANDBY"
Check Alert publish status and fetch detailed reporting back.	INFOCASTING	type="ALERT-TRACKING-DETAILS" id="1234567"

Process	API call	Notes
Alert will be ended automatically at end time (defined during publishing of an alert) but you can end it before that time by making an explicit call mentioned here.	INFOCASTING	type="ALERT" AlertData action ="END"
Retrieve all Live Alerts in the organization along with tracking summary.	INFOCASTING	Type="ALERT-LIST"

Advanced publishing

The following table lists APIs around user groups and hierarchies that may be required for advanced publishing. For example, if there are no alert templates defined within BlackBerry AtHoc for these advanced targeting settings, you can publish to any of these groups/hierarchy nodes using these APIs.

Process	API call
Retrieve Organizational Hierarchy structure and use the lineage-based node names for targeting within publishing API within Infocasting.	GET-HIERARCHY
Retrieve list of all distribution lists within given system and target any of the lists by ID or common name.	GET-DISTRIBUTION-LISTS
Retrieve list of all enabled devices within this system.	GET-DEVICES
Retrieve list of all alert folders to use for filtering the list of alert templates and alerts.	GET-ALL-CHANNELS

Using HTTP POST

To send data using HTTP POST, post the entire XML as one segment as the value of a variable named "data".

Data should be posted to the following URL:

`http://<ENS server host name>/sdk/listener/listen.asp`

For example, use the following HTML page to post data:

```
<HTML>
<HEAD>
<TITLE>Post Data to AtHoc servers</TITLE>
</HEAD>
<BODY>
<FORM name="frmMain" action="http://ens.yourcompany.com/sdk/listener/listen.asp
method="post">
<textarea cols="100" rows="20" name="data">XML Payload</textarea>
<INPUT type="submit" name="btnSubmit" value="Send feed">
</FORM>
</BODY>
</HTML>
```

BlackBerry AtHoc parses the received XML document and processes the payloads sequentially. An XML response is sent after processing is complete. Multiple payloads can be included in one XML document. The response contains a status (success or failure) for each SDK request payload.

SDK envelope request

The AtHoc SDK envelope is used to wrap SDK requests posted to the SDK listener. The following envelope format is required:

```
<AtHocSdk>
  <client>client id</client>
  <validation>
    <username>username</username>
    <password>password</password>
  </validation>
  <payload type="type" trackId="track-id">
  ...
  </payload>
  <payload type="type" trackId="track-id">
  ...
  </payload>
  ...
</AtHocSdk>
```

Note: CDATA can be used to escape special characters in names. For example: <tag><![CDATA(any text here)]></tag>

Envelope format

Element	Definition
<AtHocSdk>	The root node of the XML document
<client>	The organization ID number for which SDK requests are made
<validation>	Validation is required to ensure that data is from a valid source. The username and password specified in the XML should be a valid Management System user who has been assigned the SDK role. Usually a “service account” is generated solely to perform SDK requests. Note that both the username and password are case-sensitive. If authenticity validation fails, the data will not be accepted by the BlackBerry AtHoc system. For other types of validation, see Manage API Access and Security.
<payload>	This is a wrapper for a single SDK request. The actual SDK request is enclosed in this node. • type attribute specifies the type of SDK request • trackId optional attribute can be used to identify the SDK request and is returned in the response XML document Multiple payloads can be included in one SDK envelope and are processed serially.

The following is a sample envelope format payload:

```
<AtHocSdk>
<client>2050704</client>
<validation>
  <username>USERNAME</username>
  <password>PASSWORD</password>
</validation>
  <payload type="INFOCASTING">
<infocasting>
  <alertData>
    <scenario id="12345" />
  </alertData>
</infocasting>
</payload>
</AtHocSdk>
```

SDK envelope response

If no error occurs, there is a payload response node for every sent payload. The payload is identified by type and trackId (if provided):

```
<AtHocSdkResponse>
<payload type="type" trackId="track-id">
  <ok>
    <systemDate>utc time of server</systemDate>
    <responsePayload>
      ...
    </responsePayload>
    <warnings>
      <warning>
        <id>warning id</id>
        <description>warning description</description>
        <moreData>some more info</moreData>
      </warning>
    </warnings>
  </ok>
</payload>
<payload type="type" trackId="track-id">
  <error>
    <systemDate>time of server</systemDate>
    <id>error id</id>
    <description>error description</description>
    <moreData>more data regarding the error</moreData>
  </error>
</payload>
  ...
</AtHocSdkResponse>
```

<responsePayload> contains the data returned from the request, if any. For example, user synchronization returns the appropriate <userId> and <functions> inside the <responsePayload> node. Other requests, which do not return data, return an empty <responsePayload>.

For example:

```
<AtHocSdkResponse>
<payload type="GROUP-SETS" trackId="GR01238766">
```

```

<ok>
<systemDate>2012-02-07T19:53:09.0524618Z</systemDate>
<responsePayload/>
</ok>
</payload>
<payload type="INFOCASTING" trackId="20133455">
<error>
<systemDate>2012-02-07T19:53:09.0524618Z</systemDate>
<id>1002</id>
<description>Invalid Date</description>
</error>
</payload>
</AtHocSdkResponse>

```

If an error occurs while parsing the SDK request XML document, the following response will be returned:

```

<AtHocSdkResponse>
<error>
<systemDate>UTC time of server</systemDate>
<id>error id</id>
<description>error description</description>
<moreData>more data regarding the error</moreData>
</error>
</AtHocSdkResponse>

```

For example:

```

<AtHocSdkResponse>
<error>
<systemDate>2012-02-07T19:53:09.0524618Z</systemDate>
<id>1000</id>
<description>parsing error</description>
<moreData>invalid at the top level of the document. (line:1, position:1)</
moreData>
</error>
</AtHocSdkResponse>

```

The following general error codes are defined:

Error Code	Message	Description
1000	Parsing error	The system was not able to parse the XML. Data regarding the exact line and exact XML error appear in the <moreData> node.
1001	Invalid login	Security data did not match the data in the system
1002	Invalid IP	Invalid Source IP
9998	No request	No XML was posted to SDK listener
9999	General error	Details appear in the <moreData> node

SDK payload types

This section lists available SDK payload types.

API	Payload type	Use
User Synchronization	USER-SYNC	User synchronization with extended attributes, and static lists.
User synchronization	USER-SYNC-NEW	User synchronization.
User Details	GET-USER	Get basic user details
Feed API	INFOCASTING	Send alert message or feed content
Alert Tracking	ALERT-TRACKING-SUMMARY	Get Alert Tracking summary report
Alert Tracking	ALERT-TRACKING-DETAILS	Get Alert Tracking detail report by users and devices.
Alert Tracking	ALERT-LIST	Get a list of alerts with or without tracking numbers, filtered or unfiltered.
Get content	GET-DISTRIBUTION-LISTS	Get a list of all distribution lists that can be targeted by given user.
Get content	GET-SCENARIOS	Get a list of alert templates based on the provided search criteria.
Get content	GET-SCENARIO	Get a particular alert template.
Get content	GET-HIERARCHY	Get the organizational, distribution list, or location hierarchy.
Get content	GET-DEVICES	Get the list of devices supported by the organization.
Get Provider Details	GET-PROVIDER	Get Basic Details for the organization.
Diagnostic Log	DIAGNOSTIC-LOG	Log information for diagnostic purposes.
Set Content	HIERARCHY-SYNC	Set the organizational, distribution list, or location hierarchy.

API	Payload type	Use
Get content	GET-ALL-CHANNELS	Get a collection of all alert folders.

Deprecated Payload Types

API	Payload type	Use
Alert Tracking	ALERT-TRACKING-DEVICE-SUMMARY	Get Alert Tracking summary by device report.
C2 Alerting	GET-USER-PERSONAL-RULES	Retrieve all rules for a user.Deprecated as of V6.1.8.84.
C2 Alerting	DELETE-PERSONAL-RULE	Delete a rule.Deprecated as of V6.1.8.84.
C2 Alerting	GET-ALL-PERSONAL-RULES	Retrieve all rules for all users.Deprecated as of V6.1.8.84.
C2 Alerting	GET-PERSONAL-RULE	Retrieve an existing rule.Deprecated as of V6.1.8.84.
C2 Alerting	CREATE-PERSONAL-RULE	Create a rule for receiving alerts.Deprecated as of V6.1.8.84.
C2 Alerting	UPDATE-PERSONAL-RULE	Update an existing rule.Deprecated as of V6.1.8.84.
Delivery	SEND-PENDING-MESSAGES	Send pending user messages.Deprecated as of V6.1.8.84.
Delivery subscription	GET-SERVICE-DELIVERY-PREFERENCES	Get user delivery preferences for a feed or alert folder service.Deprecated as of V6.1.8.
Delivery subscription	SET-SERVICE-DELIVERY-PREFERENCES	Update user delivery preferences for a feed or alert folder service.Deprecated as of V6.1.8.
Delivery subscription	GET-DELIVERY-PREFERENCES	Get user delivery preferences for a all subscribed feed and alert folder services.Deprecated as of V6.1.8.
Get content	CLICK-TRACK	Indicate that a user clicked an alert message or a feed item.Deprecated as of V6.1.8.84.
Get content	GET-CONTENT	Get content of all or some dynamic services (alert folders) for a user.Deprecated as of V6.1.8.84.

API	Payload type	Use
Get content	GET-ALL-CHANNELS	Get a collection of all or some dynamic services (alert folders) for all users.Deprecated as of V6.1.8.84. Now the GET-ALL-CHANNELS returns list of alert folders for an organization.
Group-sets synchronization	GROUP-SETS	Synchronize group-sets.Deprecated as of V6.1.8.
Infocasting	PUBLISH (System Alert only)	System Alert is now deprecated starting with 6.1.8.87 and has been replaced by Enterprise Alert.
Personal services	GET-ADDRESS-FOR-NEW-SERVICE	Get available user device addresses for new personal feed or alert folder.Deprecated as of V6.1.8.84.
Personal services	UPDATE-PERSONAL-SERVICE	Update parameters of a personal feed or alert folder.Deprecated as of V6.1.8.
Personal services	CREATE-PERSONAL-SERVICE	Create a personal feed or alert folder.Deprecated as of V6.1.8.
Personal services	DELETE-PERSONAL-SERVICE	Delete a personal service.Deprecated as of V6.1.8.
Personal services	GET-PERSONAL-SERVICE	Get parameters of a personal feed or alert folder.Deprecated as of V6.1.8.
User addresses	USER-DEVICE-ADDRESS-ACTION	Create, update and delete addresses for delivery devices. Deprecated as of 6.1.8.85 R2 CP1. Use the device element in USER-SYNC.
User Channel Subscription	GET-USER-CHANNEL-SUBSCRIPTION	Retrieve a list of user subscribed folders.
User Group synchronization	USER-GROUP-SYNC	Synchronize a user's groups. Deprecated as of V6.1.8.
User groups	GET-USER-GROUPS	Get user groups.Deprecated as of V6.1.8.
User groups	SET-USER-GROUPS	Set user groups.Deprecated as of V6.1.8.

API	Payload type	Use
User Profile Synchronization	USER-SYNC-PROFILE	Synchronize a user's desktop software.Deprecated as of V6.1.8.
User subscription	SET-USER-BUTTON-SUBSCRIPTION	Set user button subscription.Deprecated as of V6.1.8.84.
User subscription	GET-USER-BUTTONS	Get user buttons.Deprecated as of V6.1.8.84.
User subscription	GET-USER-BUTTON-SUBSCRIPTION	Get user button subscription options.Deprecated as of V6.1.8.84.
Set Content	SET-USER-DEVICE-ADDRESS	Deprecated as of V6.1.8.89
User subscription	SET-USER-CHANNEL-SUBSCRIPTION	User subscriptionDeprecated as of V6.1.8.89
User subscription	GET-USER-CHANNEL-SUBSCRIPTION	User subscriptionDeprecated as of V6.1.8.89

Content APIs

This set of content APIs allows an external application to get dynamic content.

Get distribution lists

This API returns a list of distribution lists that can be targeted by the specified operator in the system.

For more details, see [SDK envelope request](#).

The payload type is GET-DISTRIBUTION-LISTS. There is no input payload for this call. Simply use the SDK Envelope with the appropriate organization ID.

The SDK response will be:

```
<responsePayload>
<distributionLists>
<systemName>Physical System Name</systemName>
<systemVersion>AtHoc Version</systemVersion>
<providerName>Organization Name</providerName>
<lists>
<list id="6808" type="STATIC">
<name>San Mateo</name>
<commonName>SAN - MATEO</commonName>
</list>
<list id="6809" type="DYNAMIC">
<name>San Francisco</name>
<commonName>SAN - FRANCISCO</commonName>
</list>
</lists>
</distributionLists>
</responsePayload>
```

Get alert templates

Retrieve a list of alert templates.

The payload type is GET - SCENARIOS.

The payload content is optional; it is used to filter the list of alert templates returned in the response.

Alert templates were previously called scenarios.

If included, it should be in the following format:

```
<scenarios>
<channel>channelCommonName</channel>
<keywords>searchString</keywords>
<availableInHomePage>Y|N</availableInHomePage>
<availableInPublisher>Y|N</availableInPublisher>
<availableForMobile>Y|N</availableForMobile>
</scenarios>
```

<SCENARIOS> Nodes

Tag	Mandatory	Description
<channel>	No	Filter the list of alert templates by alert folder.
<keywords>	No	Filter the list of alert templates by searching for the keyword in the template name, title, and body text.
<availableInHomePage>	No	Value options: Y: The returned list of alert templates includes only those that are available for publishing on the Home Page. N: The returned list of alert templates includes all alert templates.
<availableInPublisher>	Deprecated	Value options: Y: The returned list of alert templates includes only those that are available for publishing in the Quick Publisher. N: The returned list of alert templates includes only those that are not available for publishing in the Quick Publisher.
<availableForMobile>	No	Value options: Y: The returned list of alert templates includes those that are available for publishing to the Mobile app. N: The returned list of alert templates includes those that are available for the organization.

The following is a sample SDK response:

```

<scenarios>
  <scenario id="scenarioId"/>
    <name />
    <commonName />
    <alertTitle />
    <availableInHomePage />
    <availableInPublisher />
    <availableForMobile/>
    <createdBy/>
    <createdOn/>
    <updatedBy/>
    <updatedOn/>
    <metaStore />
    <channel id="">
      <name />
      <commonname />
    </channel>
  </scenario>
  ...
</scenarios>

```

GET-SCENARIOS SDK requests and responses are internationalization-enabled. For a list of supported Windows 1252 characters, see [Internationalization-supported characters](#).

Get alert template

Retrieve Details of an alert template.

The payload type is GET-SCENARIO.

The payload content should be in the following format:

```
<scenario commonName="scenarioCommonName" id="scenario Id" />
```

GET-SCENARIO SDK requests and responses are internationalization-enabled. For a list of supported Windows 1252 characters, see [Internationalization-supported characters](#).

<SCENARIO> Nodes

Tag	Mandatory	Description
<scenario>	Yes	Attributes—Either commonName or ID is required. commonName —The alert template common name. id —The alert template ID.

The SDK response is:

```
<scenarioResponse>
<scenario id="scenario id">
<name>scenario name</name>
<commonName></commonName>
<availableInHomePage></availableInHomePage>
<availableForMobile></availableForMobile>
<createdBy></createdBy>
<createdOn></createdOn>
<updatedBy></updatedBy>
<updatedOn></updatedOn>
<channel id="channel id">
<name></name>
<commonName></commonName>
</channel>
</scenario>
<userData type="USR|IP" targetAll="Y|N">
<groupCommonName>group common name</groupCommonName>
<groupCommonName isBlocked="Y" isGroupSet="N">group common name</groupCommonName>
<groupCommonName isBlocked="N" isGroupSet="Y">groupset common name</
groupCommonName>
...
<hierarchyLocation commonname="Organizational Hierarchy|Dist Lists|IP Lists" >
<path isBlocked="Y|N">/Node1/Node12/Node121/</path>
<path isBlocked="Y|N">/Node2/</path>
...
</hierarchyLocation>
<distributionList commonname="List-A" name="List A" type="STATIC|DYNAMIC|IP"
isBlocked="N|Y" />
...
```

```

<attributeCondition id="UserAttributeId" name="AttributeName"
  commonname="AttributeCommonName" >
<condition>EQUALS | NOT EQUALS | CONTAIN | DOES NOT CONTAIN | STARTS WITH | ENDS
WITH | LESS THAN | GREATER THAN | LESS THAN EQUAL | GREATER THAN EQUAL | />
<searchValues>
<value commonname="Burlingame">94010</value>
<value>94040</value>
...
</searchValues>
</attributeCondition>
...
<deviceCondition commonName="workEmail" id="2">
<condition>EQUALS</condition>
<searchValues>
<value>jsmith@acme.com</value>
</searchValues>
</deviceCondition>
</userData >
<alertData>
<title>alert title</title>
<header>alert header</header>
<body>alert body</body>
<url>url to use when clicking on alert</url>
<notifierTemplate>desktop popup template id</notifierTemplate>
<messagePriority>1</messagePriority>
<overrideUserDeliveryPrefs>Y|N</overrideUserDeliveryPrefs>
<type>Security</type>
<severity>High</severity>
<deliverySettings type="SYS|OPR|USR">
<devices>
<device commonname="DesktopPopup" targeted="Y|N" />
<device commonname="phoneWork" deliveryPriority="2" targeted="Y|N" />
...
</devices>
</deliverySettings>
<responseOptions>
<responseOption>
<responseText>I am OK</responseText>
</responseOption>
<responseOption>
<responseText>I am not OK</responseText>
</responseOption>
</responseOptions>
<metaStore>alert meta store</metaStore>
<advancedDeliverySettings>
<AlertExtensions>
<AlertExtension CommonName="Extension Common Name">
<data>
  <element id="AlertExtension id"></element>
...
</data>
</AlertExtension>
</AlertExtensions>
</advancedDeliverySettings>
</alertData>
<placeholders>
<placeholder>
<name>Place Holder name</name>
<value>Place Holder default value, if supplied</value>
</placeholder>
...

```

```
</placeholders>
</scenarioResponse>
```

Get alert template – session ID

Return the alert templates associated with a specific session ID that is generated for an authorized Personal Safety Service Operator using a mobile device.

```
<AtHocSdk>
<client>2035387</client>
<validation>
<sessionId>6727aea1-b862-4370-a5df-9ea98e90d00c</sessionId>
</validation>
<payload type="GET-SCENARIOS">
<scenarios>
<readyForPublish>Y</readyForPublish>
</scenarios>
</payload>
</AtHocSdk>
```

The example of the SDK payload response will be:

```
<AtHocSdkResponse>
  <payload type="GET-SCENARIO" trackId="">
    <ok>
      <systemDate>2012-09-26T21:24:31.9494014Z</systemDate>
      <responsePayload>
        <scenarioResponse>
          <scenario id="10212">
            <commonName>All Clear</commonName>
            <availableInHomePage>Y</availableInHomePage>
            <availableForMobile>Y</availableForMobile>
            <readyForPublish>Y</readyForPublish>
            <createdBy>first AthoC Young test</createdBy>
            <createdOn>2012-09-13T09:42:03.6</createdOn>
            <updatedBy>Sherill FB</updatedBy>
            <updatedOn>2012-09-25T15:00:11.867</updatedOn>
            <channel id="210980">
              <name>System Default</name>
              <commonName>SYSTEM-DEFAULT</commonName>
            </channel>
          </scenario>
          <userData type="USR" targetAll="N">
            <mid isBlocked="N">ATHOC\sfletch</mid>
          </userData>
          <alertData>
            <action>PUBLISH</action>
            <header>All Clear</header>
            <body>This is a test All Clear Alert</body>
            <url />
            <notifierTemplate>Default Template</notifierTemplate>
            <audioId>BlankAudio</audioId>
            <localeCode>en-US</localeCode>
            <messagePriority>1</messagePriority>
            <deliverySettings type="USR">
              <device commonname="DesktopPopup" deliveryPriority="0" />
            </deliverySettings>
            <advancedDeliverySettings>
              <AlertExtensions />
            </advancedDeliverySettings>
          </alertData>
        </scenarioResponse>
      </responsePayload>
    </ok>
  </payload type="GET-SCENARIO" trackId="">
</AtHocSdkResponse>
```

```

        <liveDate>2012-09-27T01:24:30</liveDate>
        <endDate>2012-09-27T01:24:30</endDate>
        <UTCtimestamp>Y</UTCtimestamp>
        <responseOptions />
        <consolidateMessages>N</consolidateMessages>
        <overrideUserDeliveryPrefs>N</overrideUserDeliveryPrefs>
    </alertData>
</scenarioResponse>
</responsePayload>
<warnings />
</ok>
</payload>
</AtHocSdkResponse>

```

Get alert template – session ID and alert template common name

Return the alert template associated with a specific session ID and the alert template common name that is generated for an authorized Personal Safety Service Operator using a mobile device.

```

<AtHocSdk>
<client>2035387</client>
<validation>
<sessionId>6727aea1-b862-4370-a5df-9ea98e90d00c</sessionId>
</validation>
<payload type="GET-SCENARIOS">
<scenario id='8331'>
<readyForPublish>Y</readyForPublish>
</scenarios>
</payload>
</AtHocSdk>

```

Get alert template – access token and alert template common name

Return the alert template associated with a specific access token and the alert template common name that is generated for an authorized Personal Safety Service Operator using a mobile device.

```

<AtHocSdk>
<client>2035387</client>
<validation>
<accessToken>6727aea1-b862-4370-a5df-9ea98e90d00c</accessToken>
</validation>
<payload type="GET-SCENARIOS">
<scenario id='8331'>
<readyForPublish>Y</readyForPublish>
</scenarios>
</payload>
</AtHocSdk>

```

Get hierarchy

Retrieve the organizational, distribution list, or location hierarchy.

The payload type is GET - HIERARCHY.

The payload content should be in the following format:

```
<hierarchy type="hierarchy-type" commonName="common name"
id="hierarchy_or_DL_ID" />
```

<HIERARCHY> nodes

Tag	Mandatory	Description
<hierarchy>	Yes	Attributes: either id, type, or commonName is required. Precedence rule: id, then type, and then commonName. The first attribute found in the precedence is used, the others will be ignored. • id—The ID of a hierarchy or distribution list. • type—Specifies the type of hierarchy to return; it can be blank or one of the following: • USERBASE—The complete organizational hierarchy. • DISTLIST—Only the static and dynamic distribution lists from the organizational hierarchy. • LOCATION—IP distribution lists. • commonName—The common name of a hierarchy or distribution list.

The SDK response will be:

```
<hierarchy type="USERBASE|DISTLIST|[blank]" commonName="">
<node path="/">
<providerId>1111111</providerId>
<name>hierarchy node name</name>
<commonName>hierarchy node common name</commonName>
<lineage></lineage>
<listType>TREE|DYNAMIC|STATIC|Group|GroupSet</listType>
<availableForTargeting>Y|N</availableForTargeting>
<availableForManagement>Y|N</availableForManagement>
<node path="...">
...
</node>
</hierarchy>
```

GET-HIERARCHY SDK requests and responses are internationalization-enabled. For a list of supported Windows 1252 characters, see [Internationalization-supported characters](#).

Get devices

Retrieve the list of devices supported by the organization.

The payload type is GET-DEVICES.

There is no input payload for this call. Simply use the SDK Envelope with the appropriate organization ID.

The SDK response will be:

```
<devices>
```

```

<device>
<deviceGroupName></deviceGroupName>
<deviceGroupCommonName></deviceGroupCommonName>
<deviceName></deviceName>
<deviceCommonName></deviceCommonName>
<type>PERSONAL|MASS-NOTIFICATION</type>
</device>
...
</devices>

```

The GET-DEVICES SDK requests and responses are internationalization-enabled. For a list of supported Windows 1252 characters, see [Internationalization-supported characters](#).

The specific error and warning codes are listed in the following table.

Error Code	Message	Description
1000	XML parse error.	The system was not able to parse the XML. Data regarding the exact line and exact XML error appear in the <moreData> node.
9998	Missing Request XML.	No XML was posted to SDK listener.
1111	No Scenario with the common name or ID name was found.	The Alert Template was not found for the specified organization.
4001	User not found	The operator user ID was is not valid.
4002	Either ID, type, or commonName must be provided.	Get Hierarchy API, neither id, type, nor commonName are present in the request payload.

Get alert folders

Retrieve the list of folders supported by the organization.

The payload type is GET-ALL-CHANNELS.

There is no input payload for this call. Simply use the SDK Envelope with the appropriate organization ID.

The SDK response will be:

```

<channels>
<channel id="channel Id">
<name>channel name</name>
<commonName></commonName>
<description></description>
<status>Enabled|Disabled</status>
</channel>
...
</devices>

```

GET-ALL-CHANNELS SDK requests and responses are internationalization-enabled. For a list of supported Windows 1252 characters, see [Internationalization-supported characters](#).

Get organization parameters

This API provides the most commonly used organization parameters.

Organizations were previously called Providers.

See [SDK envelope request](#) for more information.

The payload type is GET-PROVIDER. There is no input payload for this call. Use the SDK Envelope with the appropriate organization ID.

The SDK response will be:

```
<responsePayload>
<provider>
<providerId>ProviderId</providerId>
<providerName>ProviderName</providerName>
<productName>ProductName</productName>
<userAuthentication>ATUSR|WU</userAuthentication>
<homePageUrl>ProviderHomePage</homePageUrl>
<activationUrl>EndUsersActivationURL</activationUrl>
<checkUpdateInterval>CheckUpdatePollingInterval</checkUpdateInterval>
<environmentUrl>Environment URL</environmentUrl>
<logoUrl>Logo URL</logoUrl>
<clientActiveXName>ClientApplicationActiveXName</clientActiveXName>
<dateTime>
<dateFormat>MM/DD/YYYY</dateFormat>
<timeFormat>HH:MM:SS</timeFormat>
<netDateFormat>MM/dd/yyyy</netDateFormat>
<netTimeFormat>HH:mm:ss</netTimeFormat>
<timeZone>Time zone name</timeZone>
<timeZoneOffset>Offset from GMT</timeZoneOffset>
</dateTime>
</provider>
</responsePayload>
```

Publish alert API

The AtHoc Publish Alert API defines an XML format to publish alerts through BlackBerry AtHoc.

The delivery of the message is through one of the delivery devices implemented, such as email, pager, instant message, or through the AtHoc Desktop software as a desktop popup. A notification message includes a title, message body, and optional URL. In its most basic form, an alert is composed of content, user targeting, delivery devices, and scheduling. All alerts are sent based on templates in the BlackBerry AtHoc system. Nodes that are not specified fall back to the content in the alert template. Alert payloads that do not specify a template use the New Alert template.

The following topics are covered in this chapter:

- How to use the Publish Alert API
- How to publish alerts in the enterprise
- How to customize device options
- Sample payloads for various types of alerts

Alert payload definition

XML format

See [SDK envelope request](#) section for detailed formatting specifications.

The payload type is INFOCASTING.

The payload content should be in the following format:

```
<infocasting type="ALERT|ALERTNAME|ALERTSCENARIO" id="serviceId|serviceCommonName|
serviceName|scenarioCommonName|scenarioId">
  <!-- Alert Content, Delivery Devices, and other details -->
  <alertData>
    <channel>
      <commonName>alert folder common name </commonName>
    </channel>
    <scenario>
      <commonName> scenario common name </commonName>
    </scenario>
    <action>STANDBY|PUBLISH|UPDATE|APPEND|END </action>
    <audid>a unique id generated by the alert creator</audid>
    <header>alert header</header>
    <body>alert body</body>
    <url>url to use when clicking on alert</url>
    <location>{"features":[{"geometry":{"coordinates":
[[[-118.117335,34.103998],[-118.117335,33.929841],[-117.845423,33.929841],
[-117.845423,34.103998],[-118.117335,34.103998]]],"type":"Polygon","bbox":
[-118.117335,33.929841,-117.845423,34.103998]}],"properties":
{"status":"add"},"type":"Feature","bbox":
[-118.117335,33.929841,-117.845423,34.103998]}],"type":"FeatureCollection"}</
location>
    <startDate>scheduled start date</startDate>
    <liveDate>alert duration date</liveDate>
    <UTCtimestamp>Y</UTCtimestamp>
    <endDate>alert end date</endDate>
    <type>Security</type>
    <severity>High</severity>
    <responseOptions>
```

```

<responseOption>
<responseId />
<responseText>I am joining conference now</responseText>
<callBridgePhoneNumber>800-455-4555X21344</callBridgePhoneNumber>
</responseOption>
<responseOption>
<responseId />
<responseText>I cannot join the conference</responseText>
</responseOption>
</responseOptions>
<metaStore/>
<metaStoreJSON/>
<notifierTemplate>desktop popup template id</notifierTemplate>
<audioId>desktop popup audio id</audioId>
<messagePriority>1</messagePriority>
<consolidateMessages>Y|N</consolidateMessages>
<!-- Specifying Delivery Devices -->
<deliverySettings type="SYS|OPR|USR">
<device commonname="DesktopPopup" />
<device commonname="phoneWork" deliveryPriority="2" />
</deliverySettings>
<!-- Specifying Delivery Devices Extended Options -->
<advancedDeliverySettings>
<AlertExtensions>
<AlertExtension ExtensionID="id" CommonName="common name">
<data>
<element id="id"/>
</data>
</AlertExtension>
</AlertExtensions>
</advancedDeliverySettings>
<origin type="CASCADING | FEED | RECURRING | MANUAL">
<system>Alert Origin System Name</system>
<provider>Origin Provider (organization) Name</provider>
<publisher>Originating Alert Publisher</publisher>
</origin>
</alertData>
<!-- Targeting Users by various mechanisms -->
<userData type="USR|IP" targetAll="Y|N" >
<!-- Targeting individual Users -->
<id isBlocked="Y|N">user id</id>
<id isBlocked="Y|N">user id</id>
<mid isBlocked="Y|N">mapping user id</mid>
<mid isBlocked="Y|N">mapping user id</mid>
<!-- Targeting by Groups -->
<groupId>group id</groupId>
<groupId isBlocked="Y" isGroupSet="N">group id</groupId>
<groupId isBlocked="N" isGroupSet="Y">groupset id</groupId>
<groupCommonName>group common name</groupCommonName>
<groupCommonName isBlocked="Y" isGroupSet="N">group common name</groupCommonName>
<groupCommonName isBlocked="N" isGroupSet="Y">groupset common name</groupCommonName>
<!-- Targeting by Org Hierarchy/Dist List Folders/IP List Folders -->
<hierarchyLocation commonname="Organizational Hierarchy|Dist Lists|IP Lists" >
<path isBlocked="Y|N">/Node1/Node12/Node121</path>
<path isBlocked="Y|N">/Node2</path>
</hierarchyLocation>
<!-- Targeting by individual Dist Lists -->
<distributionList commonname="List-A" type="STATIC|DYNAMIC|IP" isBlocked="N|Y" />
<!-- Targeting by Advanced Query -->

```

```

<attributeCondition id="UserAttributeId" name="AttributeName" commonname="
  AttributeCommonName" >
<condition>EQUALS | NOT EQUALS | CONTAIN | DOES NOT CONTAIN | STARTS WITH | ENDS
  WITH | LESS THAN | GREATER THAN | LESS THAN EQUAL | GREATER THAN EQUAL | </
condition>
<searchValues>
<value>Burlingame</value>
<value>94040</value>
</searchValues>
</attributeCondition>
<deviceCondition commonname="workEmail" id="2">
<condition>EQUALS</condition>
<searchValues>
<value>jsmith@acme.com</value>
</searchValues>
</deviceCondition>
</userData>
  <!--Place Holders -->
<placeholders>
<placeholder>
<name>Placeholder name</name>
<value>Placeholder value</value>
</placeholder>
</placeholders>
</infocasting>

```

Mandatory nodes

<alertData> (mandatory for an alert message)

This node contains the data for an alert message, including the following alert-specific fields:

Table 1: <ALERTDATA> Tags

Tag	Mandatory	Description
<action>	No	Optional action: STANDBY, PUBLISH, UPDATE, END. Defines the action to perform. If <action> is not specified, PUBLISH is the default action. For UPDATE the <auid> node is mandatory; for END either <auid> node or <id> node is mandatory. END of an alert moves it to the Ended. STANDBY is used to put an alert in Standby mode, without publishing the alert. PUBLISH publishes a new alert, or publishes an alert in Standby (use the same <auid> value.)
<channel>	No	Element: commonName—The alert folder common name. See the rules under <scenario> in this table.

Tag	Mandatory	Description															
<scenario>	No	Element: commonName—The alert template common name. Rules: <table><tr><th colspan="2"></th><th colspan="2">View Alert Folder</th></tr><tr><th colspan="2"></th><th>Y</th><th>N</th></tr><tr><td rowspan="2">Valid Alert Template</td><td>Y</td><td>Use given alert folder. Use given alert template.</td><td>User given alert template, and use the alert folder of the alert template.</td></tr><tr><td>N</td><td>Use given alert folder, use a new alert template</td><td>Use new alert template, and use alert folder associated with this alert template.</td></tr></table>			View Alert Folder				Y	N	Valid Alert Template	Y	Use given alert folder. Use given alert template.	User given alert template, and use the alert folder of the alert template.	N	Use given alert folder, use a new alert template	Use new alert template, and use alert folder associated with this alert template.
		View Alert Folder															
		Y	N														
Valid Alert Template	Y	Use given alert folder. Use given alert template.	User given alert template, and use the alert folder of the alert template.														
	N	Use given alert folder, use a new alert template	Use new alert template, and use alert folder associated with this alert template.														
<header>	Yes	Title; max size is 200 characters, longer will be truncated. The <header> tag is internationalization-enabled.															
<body>	No	Notification body; max size is 4000 characters, longer will be truncated. The <body> tag is internationalization-enabled.															
<url>	No	The notification target URL; max size is 1024 characters															

Tag	Mandatory	Description
<audid>	No	A unique ID per organization generated by the creator of the alert. This ID can be used by the client to track or end alerts. If an incoming alert does not contain audid, the BlackBerry AtHoc will generate a unique GUID as the AUID and pass it back in the response payload.
<location>	No	This tag contains location information for an alert. The data format for this tag must be Geo Json.
<messagePriority>	No	Priority when used with deliverySettings = SYS, will be used to determine which system delivery preference to use for the alert. Priority can be a value between 1 and 3. The default value is taken from the alert scenario or alert folder.
<deliverySettings>	No	Attribute: type—possible values: SYS, OPR • SYS—(default) use system defined device delivery preferences for a given message priority. • OPR—Use the device delivery preferences specified in the subnode <devices>
<metaStore>	No	XML-based additional information for the alert.
<metaStoreJSON>	No	JSON-based additional information for the alert. It is a best practice to include a client identifier and version in this payload because there may be alerts that were not sent by the same SDK client or version of that SDK client.
<advancedDeliverySettings>	No	Advanced options for device delivery, such as templates, alternate for options (Cloud services, Mir3, giant voice, LMR, etc.)
<cap11>	No	Alert message specified in Common Alerting Protocol version 1.1 format that will be stored as is with other alert details.
<cdcPhin>	No	Alert message specified in CDC-PHIN format that will be stored as is with other alert details.

Tag	Mandatory	Description
<audioid>	No	Desktop Popup Audio ID (taken from the Audio Files manager on the Management System). Default is taken from alert folder, if not specified.
<notifierTemplate>	No	Desktop popup template ID; if no template ID is specified, the alert folder default desktop popup template is used.
<UTCtimestamp>	No	When set to Y, the alert start, live and end dates mentioned below are considered in UTC time zone. Default value is N, which uses the local time zone.
<startDate>	No	When the notification will become live (distributed to end-users). If not specified, the notification becomes live immediately. Format: yyyy-mm-ddThh:mm:ss
<liveDate>	No	Time to keep the alert message alive and display it in the alert history to end users. Cannot be earlier than startDate. If not specified, the default live date for the alert folder is used. If liveDate is specified for an existing alert along with the AUID, it updates the existing alert message. Format: yyyy-mm-ddThh:mm:ss
<endDate>	No	Expiration date and time for the alert. After the expiration time, the alert will not be available to end users in the alert history. Cannot be earlier than liveDate. If not specified, the default end date for the alert template is used. If endDate is specified for an existing alert along with the AUID, it updates the existing alert message. Format: yyyy-mm-ddThh:mm:ss

Tag	Mandatory	Description
<type>	No	Type value for the alert. Expected values include: • CBRNE • Env • Fire • Geo • Health • Infra • Met • Rescue • Safety • Security • Transport
<severity>	No	Severity value for the alert. Expected values include: • High • Moderate • Low • Informational • Unknown
<reponseOptions>	No	These options can be specified to obtain a response from end users upon message delivery. This response is tracked in the system and is available through alert tracking reports. The <responseOptions> tag is internationalization-enabled.

Tag	Mandatory	Description
<origin>	No	This node provides additional visibility into the alert being published within the BlackBerry AtHoc system. If specified, the originating details about the alert are maintained along with the alert definition, and are displayed in Sent Alerts. Origin node has following sub nodes: - Attribute: type—possible values: CASCADING, FEED, RECURRING, MANUAL (default is MANUAL) - Sub Node: system—name of the physical system where the alert originated. - Sub Node: provider—name of the organization where the alert originated. - Sub Node: publisher—name of the publisher who published this alert at the original location.
<consolidateMessages>	No	This node controls the consolidation of messages for a shared phone or email address into one message. Possible values: Y or N, Default value: Y Note that under consolidation mode, the phone messages with Pin or validate recipient requirements will not be delivered.

The date and time parameters should be specified in absolute time, using the server time zone and formatted as XML dateTime: yyyy-m-dThh:mm:ss. For example, 2003-9-10T15:7:00.

Optional nodes

<userData> (optional)

- The userData node is used for targeting content to individual users or groups of users.
- The following types of user identification can be used:

Table 2: <USERDATA> Tags

Tag	Mandatory	Description
type (attribute)	No	Values are: USR, IP USR—(Default) User based alerting. IP—IP based alerting.

Tag	Mandatory	Description
targetAll (attribute)	No	Values are: Y, N Y—Target all user base. This can be used along with blocked groups and users. N—(Default) Do not target all user base.
<Id>or<mid>	No	These identifiers can be used when a specific user needs to be alerted. ID is the internal AtHoc numeric user ID. mid is the user mapping ID (alias). Alerting a large number of individual users in this manner is not recommended for performance reasons. Distribution lists should be used in this case.
<groupId> or <groupName>	No	<i>Group ID or Group Common Name</i> for the group to be targeted. The term group is used to represent custom attributes of end users with the following attribute types: • Single PickList • Multi Picklist • CheckBox or Boolean Other optional targeting options: • isBlocked—Whether to block a certain group (Default value is targeted if this attribute is not specified) • isGroupSet—Whether the specified group is a GroupSet or a Group (Default value is Group, if this attribute is not specified)
<hierarchyLocation>	No	If an alert is targeted by hierarchy, specify the hierarchy common name (from Management System) and path to the targeted node. HierarchyLocation can also be used to target distribution list folder nodes. If more than one hierarchyLocation is specified, users belonging to any of the specified paths will be targeted.

Tag	Mandatory	Description
<distributionList>	No	If an alert is targeted to specific distribution lists (Static, Dynamic or IP), specify their common name or Id number. If more than one distributionList node is specified, users belonging to either of these lists will be targeted (inclusive).
<attributeCondition>	No	If an alert is targeted using an advanced search query on user attributes, specify the user attribute ID, common name or name and search condition and value. If the search is on pick list based attributes, the <value> node within <searchValues> can contain either the pick list value ID or common name. If more than one value is specified for one attribute, the search will be inclusive (Boolean OR) of all values within the given attribute condition. If more than one attribute condition is specified, the search will be exclusive (Boolean AND) across these conditions.
<deviceCondition>	No	Same as attributeCondition except this is used to target users based upon queries on their device addresses. For example, target all users whose email address ends with yahoo.com using the following search condition: <pre><deviceCondition commonName="homeEmail"> <condition>ENDS WITH</condition> <searchValues> <value>yahoo.com</value> </searchValues> </deviceCondition></pre>

Multiple identity criteria for a single identity type can be specified. In these cases, the alert is sent to all users who meet the criteria. Note that identity criteria types cannot be mixed. For example, individual user IDs and group IDs cannot be targeted in the same INFOCASTING call. If no userData is specified, the alert is distributed to all users specified in the alert template.

<placeholders> (optional)

- The placeholders node contains one or more Place Holder nodes.

- Place Holders in Alert text (header, body, URL, response options, and alert extensions) are searched by placeholder tag and replaced with the provided value.
- Syntax: [[Place Holder name]] or [[Place Holder name=default value]]

Table 3: <PLACEHOLDER> Tags

Tags	Mandatory	Description
Name	Yes	Name of the Place Holder. For system place holders, one of the reserved names: <i>PublishDate</i>, <i>PublishTime</i>, <i>Timezone</i>, <i>VirtualSystemName</i>, <i>OperatorFullName</i>, <i>OperatorEmail</i>, or <i>SystemName</i>. For custom place holders, the name must be unique within the organization, and cannot include square brackets - <code>[[]]</code> or names reserved for system place holders. The place holder name must match a pre-defined custom place holder name.

Tags	Mandatory	Description
Value	Yes, for custom placeholders	Replacement value for the custom place holder. For custom placeholders, the value can be one of the following types: • Text: a text string • Date: a date value that is compatible with .NET requirements¹ • Time: a time value that is compatible with .NET requirements¹ • DateTime: a datetime value compatible with .NET¹ requirements. • Single-selection: a node value that matches one of the pre-defined values for the placeholder. <value>value1</value> • Multiple-selection: one or more node values that matches one or more pre-defined placeholder values. <value>value1</value> <value>value2</value> <value>value3</value>

¹ See the .NET Date and Time format string examples on the Microsoft.com Web site.

Place holder examples

The following two examples show how you can specify place holders.

Type: Time

The following example displays a custom placeholder that specifies a time value.

```
<placeholders>
  <placeholder>
    <name>Meeting Time</name>
    <value>1:00:00</value>
  </placeholder>
  ...
</placeholders>
```

Type: multiple-selection

The following example displays a custom placeholder that specifies multiple selections from previously specified values.

```
<placeholders>
<placeholder>
<name>Colors</name>
<value>Red</value>
<value>Blue</value>
<value>Yellow</value>
</placeholder>
...
</placeholders>
```

Advanced delivery settings

These extensions usually require services or hardware from other vendors. The mandatory tags are all required when using alert extensions.

Table 4: <ALERTEXTENSIONS> Tags

Tag	Mandatory	Description
<AlertExtension>	Yes	AlertExtension node has the following attributes: • Attribute ExtensionId—Valid values depend on what the system has been configured with. • Attribute CommonName—The common name of the alert extension.
<data>	Yes	Container for the element extension properties.
<element>	Yes	Elements describe extension properties. Attribute id – distinguishes one element from another. See Sample Extensions below.

Sample extensions

```
<advancedDeliverySettings>
<AlertExtensions>
<!--Example Telephony Extension-->
<AlertExtension ExtensionID="" CommonName="TELEPHONY-EXTENSIONS">
<data>
<element id="AlertExtension.DeviceGroup.Telephony.DeliveryType">Custom</element>
<element
  id="AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.StopCalling.Criteria">Answered
element>
<element
  id="AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.ValidateRecipients"
>
```

```

<element
  id="AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.RequirePIN" /
>
<element
  id="AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.ReplayMessage" /
>
<element
  id="AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.LeaveMessage" /
>
<element
  id="AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.LeaveCallback" /
>
<element id="AlertExtension.DeviceGroup.Telephony.ContactCycle.Count">1</element>
<element id="AlertExtension.DeviceGroup.Telephony.ContactCycle.Delay">00:05</
element>
</data>
</AlertExtension>
<!--Example GiantVoice Extension-->
<AlertExtension ExtensionID="" CommonName="GV-EXTENSIONS">
<data>
<element
  id="AlertExtension.DeviceGroup.LandMobileRadio.ContentSource">TextToSpeech</
element>
<element
  id="AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.Audio.AudioID" />
<element id="AlertExtension.DeviceGroup.GiantVoice.ContentSource.Key.KeyID" />
<element
  id="AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.TextToSpeech.TTSSource">Messag
element>
<element
  id="AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.TextToSpeech.TTSSource.CustomT
>
</data>
</AlertExtension>
<!--Example Cisco Extension -->
<AlertExtension ExtensionID="" CommonName="CISCO-UAP-EXTENSIONS">
<data>
<element id="AlertExtension.DeviceGroup.UCM.Visual.DisplayImage">CustomURL</
element>
<element id="AlertExtension.DeviceGroup.UCM.Visual.DisplayImage.Image.File" />
<element
  id="AlertExtension.DeviceGroup.UCM.Visual.DisplayImage.CustomURL.Path">http://
athoc.com/logo.gif</element>
<element id="AlertExtension.DeviceGroup.UCM.Ringtone.Source">Custom</element>
<element id="AlertExtension.DeviceGroup.UCM.Ringtone.Source.Custom.File">|
ClockShop.raw</element>
<element id="AlertExtension.DeviceGroup.UCM.Audio.Source">Body</element>
<element id="AlertExtension.DeviceGroup.UCM.Audio.Source.Custom.Value" />
<element id="AlertExtension.DeviceGroup.UCM.Audio.RepeatAudio">on</element>
<element
  id="AlertExtension.DeviceGroup.UCM.Audio.RepeatAudio.RepeatAudioValue">3</
element>
</data>
</AlertExtension>
<!--Example Email Extension -->
<AlertExtension ExtensionID="1010" CommonName="SMTP-EXTENSIONS">
<data>
<element id="AlertExtension.DeviceGroup.Template.TemplateType">Custom|Default</
element>
<element id="AlertExtension.DeviceGroup.Template.TemplateType.Custom.Id"><template
ID></element>

```

```

<element id="AlertExtension.DeviceGroup.Email.Source">HeaderBody|Custom</element>
<element id="AlertExtension.DeviceGroup.Email.Source.Custom.Title">Body Text</
element>
<element id="AlertExtension.DeviceGroup.Email.Source.Custom.Body">Custom body
text</element>
</data>
</AlertExtension>
</AlertExtensions>
</advancedDeliverySettings>

```

ExtensionId: 1001, CommonName: TELEPHONY-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType

Values:

- Informational
- emergency (default)
- Secure
- Custom

Element ID: AlertExtension.DeviceGroup.Telephony.ContactCycle.Delay

Values:

- hh:mm (default = 00:10)

Element ID: AlertExtension.DeviceGroup.Telephony.ContactCycle.Count

Values:

- Numeric (default = 1)

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.StopCalling.Criteria

Values:

- Listened
- Acknowledged
- Voice Mail

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.RequirePIN

Values:

- On
- Empty

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.validateRecipients

Values:

- On
- Empty

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.ReplayMessage

Values:

- On
- Empty

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.LeaveMessage

- On

- Empty

Element ID: AlertExtension.DeviceGroup.Telephony.DeliveryType.Custom.CallOption.LeaveCallback

Values:

- On
- Empty

ExtensionId: 1002, CommonName: GV-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource

Values:

- TextToSpeech
- Audio
- Key

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.Audio.AudioID

Values (Depends on ContentSource):

- TextToSpeech: empty
- Audio: the full url to csi/getAudio.asp?id=audio Id

Element ID: AlertExtension.DeviceGroup.GiantVoice.ContentSource.Key.KeyID

Values:

- Numeric (predefined GV Key Id)
- Empty

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.TextToSpeech.TTSSource

Values:

- Full
- Header
- Body
- Custom

Element

ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.TextToSpeech.TTSSource.CustomText.Value

Values:

- Unstructured text

ExtensionId: 1003, CommonName: LMR-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource

Values:

- TextToSpeech
- Audio
- Key

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.Audio.AudioID

Valued (depends on ContentSouce):

- TextToSpeech: empty
- Audio: the full url to csi/getAudio.asp?id=audio Id

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.TextToSpeech.TTSSource

Values:

- Full
- Header
- Body
- Custom

Element ID: AlertExtension.DeviceGroup.LandMobileRadio.ContentSource.TextToSpeech.TTSSource.Custom.Value

Values:

- Unstructured text

ExtensionId: 1004, CommonName: CISCO-UAP-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.UCM.Audio.Source

Values:

- None
- HeaderBody
- Header
- Body
- Custom

Element ID: AlertExtension.DeviceGroup.UCM.Audio.Source.Custom.Value

Values:

- unstructured text, minimum: 10 characters, maximum: 3000 characters

Element ID: AlertExtension.DeviceGroup.UCM.Audio.RepeatAudio

Values:

- on
- empty

Element ID: AlertExtension.DeviceGroup.UCM.Audio.RepeatAudio.RepeatAudioValue

Values: Numeric (Default =3)

Element ID: AlertExtension.DeviceGroup.UCM.Ringtone.Source

Values:

- None
- empty (use .Custom.File)

Element ID: AlertExtension.DeviceGroup.UCM.Ringtone.Source.Custom.File

Values: —

Element ID: AlertExtension.DeviceGroup.UCM.Visual.DisplayImage

Values:

- Image
- Camera
- CustomURL

Element ID: AlertExtension.DeviceGroup.UCM.Visual.DisplayImage.Image.File

Values: —

Element ID: AlertExtension.DeviceGroup.UCM.Visual.DisplayImage.CustomURL.Path

Values: —

ExtensionId: 1010, CommonName: EMAIL-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.Template.TemplateType

Values:

- Custom —The Custom delivery template
- Default—Uses the Other default email template.

Element ID: AlertExtension.DeviceGroup.Template.TemplateType.Custom.Id

Values:

- Delivery template ID is required.
- When the TemplateType is Default, specify ?
- When TemplateType is Custom, enter the templateId of your choice.

Element ID: AlertExtension.DeviceGroup.Email.Source

Values:

- HeaderBody—Use the alert text.
- Custom—Use custom text.

Element ID: AlertExtension.DeviceGroup.Email.Source.Custom.Title

Values: Custom title text just for email.

Element ID: AlertExtension.DeviceGroup.Email.Source.Custom.Body

Values: Custom title text just for email.

ExtensionId: 1028, CommonName: SMS-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.ShortMessageService.Source

Values:

- Header
- Body
- HeaderBody(Default)
- Custom

Element ID: AlertExtension.DeviceGroup.ShortMessageService.Source.Custom.Value

Values: Unstructured text, minimum 10 characters, maximum 3000 characters.

ExtensionId: 1029, CommonName: SMTP-EXTENSIONS

Element ID: AlertExtension.DeviceGroup.Email.Source

Values: Custom

Element ID: AlertExtension.DeviceGroup.Email.Source.Custom.Title

Values: Unstructured text, minimum 10 characters, maximum 3000 characters.

Element ID: AlertExtension.DeviceGroup.Email.Source.Custom.Body

Values: Unstructured text, minimum 10 characters, maximum 3000 characters.

Publish alert response

The response format is defined in the SDK Envelope Response section of this guide. The response payload will also contain the Alert ID for an alert created within the BlackBerry AtHoc system, in the following format:

```
<responsePayload>
<alert>
<id>1036472</id>
<aid>8660A541-D666-4F7D-BFA7-6A1E410BF5A8</aid>
</alert>
</responsePayload>
```

The specific error and warning codes are listed in the following tables.

Error Code	Message	Description
1000	Missing Infocasting XML	Payload request infocasting action is missing.
1001	ID missing in Infocasting node	The ID in the Infocasting node or the common alert template name has not been provided.
1002	This service is not slotted in the button.	Service should be slotted in the button to be available for publishing.
1003	Unknown Infocasting type	Invalid infocasting type.
1101	Invalid AlertName	AlertName was not found.
1103	Missing AUID	AUID is not available. AUID is necessary for END and UPDATE action.
1104	No Alert Found for AUID	AUID is invalid as no alert exist for the AUID.
1105	Invalid/Unknown Action	Action is not recognized. Valid actions are PUBLISH, UPDATE, and END.
1109	Error creating alert	Alert was not created due to some error. The <moreData> node provides additional explanation.
1111	Invalid AlertScenarioName. Could not resolve into Scenario ID	The alert template was not found for the specified organization.
1112	Missing AlertScenarioName	The alert template was not found as part of the payload.
1113	Group Attribute defined incorrectly	The attribute "isBlocked" is not defined correctly for target groups of the alert.

Error Code	Message	Description
1114	isBlocked attribute has wrong value	The group attribute "isBlocked" can only have "Y" or "N".
1201	Invalid ServiceId	Service ID passed in the id attribute is not valid.
1202	Invalid FeedName	Feed Name passed in the id attribute is not valid.
1203	Web Listener is not enabled for this feed	Web Listener needs to be enabled for a feed to allow it be published using SDK. Value is controlled from the management system.
1204	Failed to Update Raw Content	Raw Content could not be updated for managed feed.
1205	Start Date occurs in the past.	The start date specified occurs in the past.
1206	Live Date occurs before Start Date.	Live date cannot occur before the alert publish date.
1207	Expiration Date occurs before Start Date.	Expiration date cannot occur before the alert publish date.
1208	Expiration Date occurs before Live Date.	Expiration date cannot occur before the Live date.
1209	No Service Information found	No Service Information was found for the given ID.
2301	Failed to Create Personal Alert	Personal alert was not created. The <moreData> node provides additional explanation.
2302	No User Information found for Personal Alert	The Link ID did not return any information about the owner of the service.
2303	The user of this Personal Service does not belong to the calling Provider	The organization making the SDK call does not have ownership over the service it is trying to publish.
2401	Failed to Update Gdum Content	Failed to update Raw Content for personal feed. The <moreData> node provides additional explanation.

Error Code	Message	Description
2402	Failed to Generate Result	Failed to process the raw content. The <moreData> node provides additional explanation.

Warning Code	Message	Description
2101	Update failed	Update fails if Alert has already ended or expired. The <moreData> node provides additional explanation.
2201	Content is empty	Incoming Content was empty and the feed has been updated with it.
2202	Failed to Update Raw Content for User	Raw Content was not updated for a user for personalized managed feeds. The <moreData> node provides additional explanation.
2203	Could not resolve MID	The MID could not be resolved into a userId for personalized managed feeds.
2204	Raw Content not updated as it was same as before	Raw Content is only updated if there is a change from the previous update.
2205	Long Data	Data too long, truncation will be performed.

Publish enterprise alerts

Note: If you are currently using the System Alert specification in your current payloads, we recommend using the Publish Alert specification upon upgrade to this version. For backward compatibility, the system does not throw any errors if system alert specifications are still present. However these settings will be ignored by the system.

Publishing an Enterprise alert extends the alert publish function. This extension publishes one alert that targets users from multiple sub organizations with default setting from the originating organization (including alert template and account.) To publish an Enterprise alert, complete the following tasks:

- **Enterprise Organization Setup:** configure an organization as the Enterprise organization, within the BlackBerry AtHoc system. This enterprise organization has the common user attributes that are used across all sub organizations.
- **SDK user:** create a user account on the Enterprise organization that has the SDK role. The user base can be either unrestricted (access to all sub organization users) or restricted by user attribute or sub organization access.

After you have completed the set up, provide the Enterprise organization ID and the SDK user credentials in the SDK envelope for Publishing. When you submit your publishing payload with these settings, the payload

creates an Enterprise alert within the Enterprise organization that targets users across other sub organizations, as specified by SDK user's user base.

Important: Enterprise publishing targets recipients in all organizations.

- Published alerts always go out to the Enterprise organization and all sub organizations.
- You cannot target a specific sub organization. All organizations in the Enterprise are targeted, as long as they are in the user base or the operator.

Alert tracking API

The following topics are covered in this chapter:

- Alert list
- Summary report
- Detail Report by users and devices

Alert list

Retrieve a list of alerts. See [SDK envelope request](#) for more information.

The payload type is INFOCASTING .

The payload content should be in the following format:

```
<infocasting type="ALERT-LIST" id="1234567"
  auid="1111111-1111-1111-1111-111111111111">
  <alertStatus>alert-status</alertStatus>
  <channel>alert-folder</channel>
  <publishedBy>username</publishedBy>
  <startDate>Alert start date (search range start) </startDate>
  <endDate> Alert start date (search range end) </endDate>
  <includeTracking>Y|N</includeTracking>
  <includeMetastore>Y|N</includeMetastore>
</infocasting>
```

Table 5: <INFOCASTING> Tags

Tag	Mandatory	Description
<infocasting>	Yes	Root node of the payload content. Attributes: • type—Required, the type of Alert Tracking. Must be ALERT-LIST. • id—Optional, the alert Id. If id is present, auid and the other request payload nodes are ignored, except includeTracking. • auid—Optional, the system-generated value. If present, the other request payload nodes are ignored, except includeTracking. See Publish alert API for additional uses for this node.
<alertStatus>	No	One or more of the following status types, comma-delimited (not case-sensitive): • live • standby • scheduled • ended

Tag	Mandatory	Description
<channel>	No	Numeric alert folder that may be a comma-delimited list of IDs. If this tag is not present, alerts from all folders are returned.
<publishedBy>	No	Username of the user who published the alert(s).
<startDate>	No	Start of search date range: include alerts in which the date and time that the notification became live (distributed to end-users) is after startDate. Format: yyyy/mm/ddThh:mm:ss
<endDate>	No	End of search date range: include alerts in which the date and time that the notification became live (distributed to end-users) is before endDate. If present, must be later than search range start time. Format: yyyy/mm/ddThh:mm:ss
<includeTracking>	No	If present, tracking numbers are included in response.
<includeMetastore>	No	If Y is present, the metastoreJSON property is returned for each alert in the list.

The SDK response will be:

```

<alerts>
<alert id="alert id" auid="auid">
<title>alert title</title>
<body>alert body</body>
<url>Target URL</url>
<responseOptions>
<responseOption>
<responseId></responseId>
<responseText>ok</responseText>
</responseOption>
.
.
.
</responseOptions>
<channel>folder name</channel>
<startDate></startDate>
<endDate></endDate>
<publishedBy>username</publishedBy>
<alertStatus>Standby|Scheduled|Live|ENDED</alertStatus>
<type>Other</type>
<severity> Unknown</severity>
<!--Tracking numbers are included when includeTracking = Y in request-->
<targeted>123</targeted>
<sent>123</sent>

```

```

<received>123</received>
<acknowledged>111</acknowledged>
<noResponse>12</noResponse>
<!--Metastore in JSON format is included when includeMetastore=Y in request-->
    <metaStoreJSON>
        {
            "pushbuttonAlpha": {
                "deliveryType":"SDKAlert",
                "room":"3509",
                "room occupant":"John Smith"
            }
        }
    </metaStoreJSON>
</alert>
.
.
.
</alerts>

```

Summary report

Retrieves summary tracking data for a given alert.

Note: If dependents are enabled, the summary report includes consolidated information for sponsors and dependents.

For more information, see the [SDK envelope request](#).

The payload type is ALERT - TRACKING - SUMMARY .

The payload content should be in the following format:

```

<alerts>
<alert withDistributionLists="true|false" withDevices="true|false"
  withHierarchy="true|false">alertId</alert>
</alerts>

```

Table 6: <INFOCASTING> Tags

Tag	Mandatory	Description
<alerts>	Yes	Root node of the payload content.
<alert>	Yes	Specifies the alert ID for the report. Attributes: withDistributionLists—optional, true or false. When true, distribution lists are included in the response. withDevices—optional, true or false. When true, devices that were published to are included in the response. withHierarchy—optional, true or false. When true, hierarchies that were published to are included in the response.

The SDK response will be:

```
<AtHocSdkResponse>
<payload type="ALERT-TRACKING-SUMMARY" trackId="">
<ok>
<systemDate>yyyy-mm-ddThh:mm:ss.msZ</systemDate>
<responsePayload>
<report>
<alert id="alertId" name="">
<ack>1</ack>
<noResponse>0</noResponse>
<recv>1</recv>
<sent>1</sent>
<targeted>1</targeted>
<responses>
<response index="0">1</response>
<response index="1">0</response>
</responses><!-- If withDistributionLists="true", the distributionLists node is
    included in the response -->
<distributionLists>
<distributionList id="list ID" name="list name">
<ack>1</ack>
<noResponse>0</noResponse>
<recv>1</recv>
<sent>1</sent>
<targeted>1</targeted>
<responses>
<response index="0">1</response>
<response index="1">0</response>
</responses>
</distributionList>
. . .
</distributionLists><!-- if withDevices="true", the devices node is included in
    the response -->
<devices>
<device name="device name" id="device id">
<ack>1</ack>
<noResponse>0</noResponse>
<recv>1</recv>
<sent>1</sent>
<targeted>1</targeted>
<responses>
<response index="0">1</response>
<response index="1">0</response>
</responses>
</device>
. . .
</devices><!-- if withHierarchy="true", the hierarchies node is included in the
    response -->
<hierarchies>
<hierarchy id="123456789" name="hierarchy_name">
<ack>1</ack>
<noResponse>1</noResponse>
<recv>2</recv>
<sent>2</sent>
<targeted>2</targeted>
<responses />
</hierarchy>
. . .
</hierarchies>
```

```

</alert>
.
.
.
</report>
</responsePayload>
<warnings />
</ok>
</payload>
</AtHocSdkResponse>

```

Detailed report by users and devices

Retrieve detailed tracking data for a given alert, broken down by users and devices. See [SDK envelope request](#) for more information.

Note: If dependents are enabled, the detailed report includes information for sponsors and dependents.

The payload type is INFOCASTING.

The payload content should be in the following format:

```

<infocasting type="ALERT-TRACKING-DETAILS" id="1234567"
  auid="1111111-1111-1111-1111-111111111111">
  <startRowIndex>1</startRowIndex>
</infocasting>

```

Table 7: <INFOCASTING> tags

Tag	Mandatory	Description
<infocasting>	Yes	Root node of the payload content. Attributes: either id or auid is required. type—(Required) The type of Alert Tracking. Must be ALERT-TRACKING-DETAILS. id—The alert Id. Either id or auid is required. If id is present, auid is ignored. auid—The system-generated value.
<startRowIndex>	No	Default: 1 (the report will start with the first record). When the report contains more than 1000 records, it is returned in pages of 1000 records. To request the next page of records, use <endRowIndex> from the response, increment by 1, and use that for <startRowIndex>. RowIndex is automatically generated by system. It is not contiguous. You keep calling next page until the response contains 0 tracking events (<recordCount> node will contain value 0).

The SDK response will be:

```

<report>
<endRowIndex>12345</endRowIndex>

```

```
<recordCount>123</recordCount>
<events>
<event type="Sent" userid="1234567" username="name" displayName="display name"
deviceName="Desktop Popup" address="device address" timestamp="4/27/2017
12:00:00 AM" messageText="" rowIndex="12345" />
<event type="Sent" userid="8912345" username="name" displayName="display name"
deviceName="Desktop Popup" address="device address" timestamp="4/27/2010
12:00:00 AM" messageText="" rowIndex="12349" />
.
.
.
</events>
</report>
```

Notes

- `username` is same as `mid`.
- `rowIndex` is system-generated record number. It is not contiguous for same alert.
- `Timestamp` is UTC time.
- `endRowIndex` is the Max row index in response.
- `recordCount` is the total record number for the selected alert, regardless of how many are returned in the current response.

Distribution lists synchronization

BlackBerry AtHoc can synchronize distribution lists with an external source. This allows an external system to manage distribution lists and its members, and update changes in BlackBerry AtHoc.

BlackBerry AtHoc currently supports only Static Distribution Lists for synchronization.

Distribution lists definition XML

See the SDK Envelope format specification in this guide for more information.

The payload type is DISTRIBUTION-LISTS-SYNC.

The payload content should be in the following format:

```
<distributionListSync>
<syncSource>unique-ID-of-source</syncSource>
<operator id="operator-id" >operator-username</operator>
<isFullSync>Y|N</isFullSync>
<newListDefaultFolder>list-folder-location</newListDefaultFolder>
<distributionLists>
<list type="STATIC" id="list-id" commonName="list-common-name" >
<name>List-Name</name>
<description>List-Description</description>
<folder>list-folder-location</folder>
<members>
<user mid="member-user-mid" />
<user mid="member-user-mid" />
.
.
<list commonName="nested-member-list-common-name" />
<list commonName="nested-member-list-common-name" />
.
.
</members>
</list>
.
.
</distributionLists>
</distributionListSync>
```

Mandatory nodes

Node	Definition
<syncSource>	Sync Source is predefined within the BlackBerry AtHoc system. If the given source is not found within the BlackBerry AtHoc system, import fails and an error occurs. Build-in sync source includes: AD,ROSTER,GROUP-SYNC
<list>	This node represents a single list that is updated within the BlackBerry AtHoc system.

Node	Definition
<id> or <commonName>	One of these attribute nodes is mandatory for identifying the list uniquely. commonName contains the unique ID of the list within source system. Id refers to unique ID of the list within BlackBerry AtHoc system (can be specified for existing lists during updates).
<name>	Mandatory for creating new lists. It is used to check the list's existence in the BlackBerry AtHoc system. If <name> is not specified for new lists, import of the particular list fails.

Optional nodes

Node	Definition
<type>	This attribute node is optional and by default it is set to STATIC.
<operator>	This node can contain the operator user name or ID within the BlackBerry AtHoc system. If specified, this operator is granted full permissions (targeting and management) to imported distribution lists. Otherwise, the SDK user is the default owner of these lists. If an invalid value is specified, an error occurs and the import of lists is aborted.
<isFullSync>	The default is N (No). When specified as Y (Yes), import performs a full sync on lists. The lists from BlackBerry AtHoc are removed if they no longer exist in the source. When specified as N (No), import updates existing lists or creates new ones. However, missing lists from the source are not removed from the BlackBerry AtHoc system.
<newListDefaultFolder>	This node specifies the Distribution List folders location within the BlackBerry AtHoc system as the slotting folder for newly created lists. This node is not used when updating existing lists. If unspecified, the default folder location for new lists is the root folder. If an invalid value is specified, an error occurs and import of all lists is aborted.
<description>	If the description is not specified, the description for existing lists does not change. If a description is specified, it applies to all existing lists.
<folder>	If not specified, new lists are slotted in the default location. The location of existing lists is unaffected. If specified, the new location applies to all new and existing lists. If the specified folder path is invalid, an error occurs and import of the particular list is ignored.

Node	Definition
<members>	If not specified, existing members in existing lists are unaffected. If specified with no members defined inside, all members are removed from existing lists. If members are specified, they are added as members of a given list. • <user mid="user-mid" /> This node contains the mid of the end user who is a member of the existing list. • <list commonName="Group2" /> This node contains the common name of the existing list within the BlackBerry AtHoc system that is a member of the existing list. No circular nesting in lists is allowed in BlackBerry AtHoc and if this is violated, a warning is issued and the nesting of offending list is ignored.

Distribution lists synchronization response

The response format is as follows:

```

<output>
<distributionLists>
<!-- For each input list we will create corresponding output xml
Along with list information, we will create a new attribute called result
and generate a value with "OK" | "ERROR" | "WARNING" -->
<list type="STATIC" id="id" commonName="common name" result="ERROR | OK | WARNING
>
<!-- For new list name tag will be created -->
<name>Group1</name>
<!-- if new list and name is not provided we will generate error
<!-- Here is the sample error no and err desc -->
<Error>err no</Error>
<description>err desc</description>
<!-- If members are provided, if there is any invalid member we will create
invalid member list -->
<invalidMembers>
<!-- Following members were not found within target system and ignored -->
<user mid="user id" />
<list type="STATIC" id="" commonName="common name" />
</invalidMembers>
</list>
.
.
<list type="STATIC" id="id" commonName="common name" result="WARNING" >
<name>Group2</name>
<invalidMembers>
<user mid="user-mid" />
<list type="STATIC" id="" commonName="common name" >This list creates a circular
reference via nesting, cannot be added as child list.</list>
</invalidMembers>
</list>
</distributionLists>

```

</output>

The specific error and warning codes are:

Error code	Error description
4101	Sync Source is Mandatory
4105	Invalid Sync Source
4102	New List default folder location is invalid
4106	List common name already exist; please provide new unique List common name
4103	List Name mandatory for new List
4104	List folder not valid

User synchronization

User synchronization is a set of APIs that enables the following:

- Mapping of BlackBerry AtHoc profiles to user identities in an external system
- Updating attributes in BlackBerry AtHoc profiles
- Delegation of user authentication to an external authentication system

Overview

BlackBerry AtHoc manages configuration profiles for all users; a unique ID, UID, identifies every user. The UID is assigned internally by ENS. The different types of users are:

- **Non registered users (guests or signed-out users):** These user profiles are created upon activation of a desktop client configured for manual sign in.
- **Registered users:** Users who are identified by an authentication scheme. These profiles can roam; once the user is positively authenticated, the correct profile can be fetched. Only registered users can personalize their profile.

BlackBerry AtHoc provides its own registration and authentication mechanism (based on username / password).

In many corporate environments where BlackBerry AtHoc is deployed, this is not a preferred practice because a central user directory and authentication scheme is already implemented. In these environments, there should be a way to map corporate users to BlackBerry AtHoc profiles and to authenticate users using a central authentication mechanism.

Establishing a mapping between corporate users and BlackBerry AtHoc profiles is key to further integration with external systems. For example, once such a mapping is achieved, it is possible to:

- Synchronize user permissions and group membership from external systems (such as LDAP or corporate portals) and BlackBerry AtHoc.
- Post alerts and notifications to specific users.
- Integrate personalized configuration and data from corporate systems in the BlackBerry AtHoc profile.
- Provide user-level reporting of usage and notification distribution.

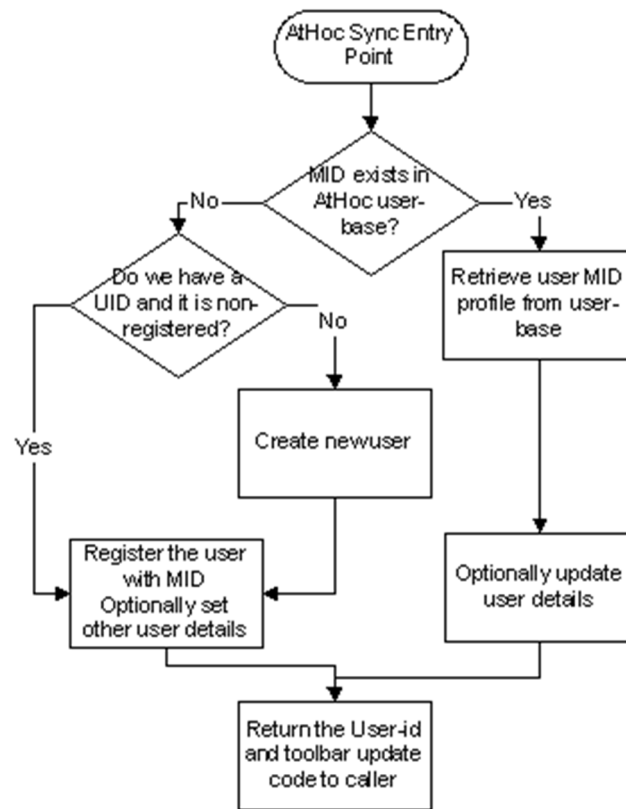
The following objectives are addressed by BlackBerry AtHoc user synchronization:

- Establish and maintain a common unique identifier per user; the Mapping ID (MID) is used to map the corporate system user identifier and the BlackBerry AtHoc UID.
- Perform sign-on and registration functions in one system only – the corporate system; no need to provide BlackBerry AtHoc with sensitive information about users. (There is an option to pass first name, last name and email as part of the user synchronization so that BlackBerry AtHoc will be able to send email alerts to end-users. This data is collected by BlackBerry AtHoc during user registration, if it is done by BlackBerry AtHoc.)
- Provide a sign-on gateway for BlackBerry AtHoc; sign-on and user authentication continue to take place in the corporate system; only the results with the appropriate MIDs are transferred to BlackBerry AtHoc.
- Optionally enable a common sign-in to the desktop software when a BlackBerry AtHoc user signs in to the corporate system.

Note: User synchronization cannot be configured in parallel with Windows user authentication because BlackBerry AtHoc does not support mixing two methods to authenticate a user. Windows user authentication is a special case of user synchronization and is described at the end of this chapter.

User synchronization process

Flow description



The most basic action is User Synchronization.

In this action, BlackBerry AtHoc obtains a Mapping ID (MID) from the client system, together with some additional optional parameters.

Using the MID, the BlackBerry AtHoc system determines whether to:

- Retrieve an already existing user profile (matching the received MID); optionally, update it with passed user details (user names, email)
- Create a new user and register it with the MID
- If a desktop software user-id is passed to BlackBerry AtHoc and is non-registered, BlackBerry AtHoc registers this user with the received MID; if additional user details are passed, they are registered as well
- After the appropriate action is determined and completed, BlackBerry AtHoc returns a proper response to the caller, together with a code snippet on how to update the desktop software, if needed.
- The user-synchronization is implemented as a server-to-server call. It is the responsibility of the caller to perform the desktop software update (using client-side script) if needed.

Updating the user profile

Frequently after user synchronization, there is a need to set or update user profiles. The "User Group Membership Synchronization" section of this guide describes in detail how to configure the synchronization.

In addition to the session initiation synchronization activity (as described in Chapter 5), it is possible to update a user profile upon initial user synchronization. For example, set user group membership. To do this, post the user group membership to BlackBerry AtHoc from the integrated system following successful user synchronization.

User synchronization

The following sections describe how to add, update, or delete synchronized user accounts.

User synchronization XML

See [SDK envelope request](#) for more information.

The payload type is USER-SYNC.

To **create** a new user or to **update** an existing user, use the following payload format:

```
<userSynchronization>
<mid>mid</mid>
<userId>user-id</userId>
<firstName>user-first-name</firstName>
<lastName>user-last-name</lastName>
<displayName>user-display-name</displayName>
<email>user-email</email>
<enabled>Y|N</enabled>
<customFields>
<field id="attributeId">value</field>
<field commonName="attribute-common-name">value</field>
...<field id="12798" commonName="AB">No</field>
</customFields>
<devices>
<device id="201" commonName="workEmail">jsmith@acme.com</device>
<device commonName="workPhone">(450) 555-111</device>
<device commonName="homePhone">(405) 777-2222</device>
</devices>
<noFunctionScript />
<syncExistingUserOnly />
</userSynchronization>
```

Note: Use either<field id... or<field commonName..., not both. The payload fails if both are used.

To **delete** an existing user, use the following format for the payload content:

```
<userSynchronization>
<mid>mid</mid>
<userId>user-id</userId>
<delete type="DEL-WO-PII|DEL-W-PII|DEL-FULL" />
</userSynchronization>
```

Note: DEL-WO-PII means to delete the user without removing the PII (Personally Identifiable Information.) DEL-W-PII anonymizes all the user information including the following nodes: username, name fields, and contact details. DEL-FULL deletes the complete user record from the system including all tracking information for the user.

Mandatory nodes

<mid>

<mid> is a mapping-id to uniquely identify the user between the systems. The MID value can be any alphanumeric string between 2 and 80 characters

Optional nodes

Node	Definition
<userId>	<userId> is an existing BlackBerry AtHoc user identity. If this node exists, the user ID included in the node is registered with the included MID, overwriting the existing user-ID and mapping-ID assignment.
<firstName> <lastName> <displayName>	These nodes can set the user first name, last name, display name in the BlackBerry AtHoc system. If the user exists and is registered, the included values are set for the user, overwriting all existing values.
<enabled>	Setting this node to N disables an end-user. Disabled end-users do not receive any alerts. Setting this node to Y re-enables the end-user.
<noFunctionScript />	If this node is present, the JavaScript code snippet to instantiate AtHoc desktop software ActiveX Object is not returned. This can improve performance for large batches of USER-SYNC operations.
<syncExistingUserOnly />	If this node is present, the user attributes are updated only if the user already exists. However, no new users are created.
<customFields>	This node contains all the user attributes and their values. Please note that these attributes must exist in the BlackBerry AtHocsystem before the SDK call is made. This node can also perform Static List membership updates and hierarchy updates. Beginning with release 6.1.8.78 it is recommended to set check box fields to Yes or No. (For backward compatibility, checkbox fields will continue to support values of 1 or 0.) The custom field identifier can be specified using id or commonName attribute. Some examples are mentioned below.
<devices>	This node contains all the user contact details. Please note that these devices must be enabled for User's organization before the SDK call is made. The device identifier can be specified using id or commonName attribute.

Static list membership updates

Add a user to a list with the common name "List-A":

```
<customFields>
<field commonName="List-A">1</field>
</customFields>
```

Remove a user from a list with the common name "List-A":

```
<customFields>
<field commonName="List-A">0</field>
</customFields>
```

Organizational Hierarchy User Update: Set the user location to the value "/Node1/Node12/Node121/" in an organizational hierarchy with the common name "Organizational Hierarchy":

```
<customFields>
<field commonName="Organizational Hierarchy"> /Node1/Node12/Node121/</field>
</customFields>
```

Note: The id attribute can be used instead of commonName.

User synchronization response

After successful synchronization, nodes are returned in the <ok> node. The <noFunctionScript /> node changes the content that is returned in the response.

Response without the <noFunctionScript> node

The following response is returned when the <noFunctionScript /> node is not present:

```
<AtHocSdkResponse>
  <payload type="USER-SYNC" trackId="">
    <ok>
      <systemDate>time of server</systemDate>
      <responsePayload>
        <userId>user-id</userId>
        <mid>mid</mid>
        <token>token</token>
        <functions>
          <![CDATA[ Returned JavaScript code
        ]]>
        </functions>
      </responsePayload>
      <warnings>
        <warning>
          <![CDATA[ Returned warnings
        ]]>
        </warning>
      </warnings>
    </ok>
  </payload>
</AthocSdkResponse>
```

For example:

```
<ok>
<systemDate>10/8/2003 5:32:30 pm</systemDate>
<responsePayload>
<userId>555555</userId>
<mid>213765137265</mid>
<token>abcdabcd</token>
<functions>
<![CDATA[
if (document.all) {
var xATHOC = new ActiveXObject("GStlbar.GShelper");
if ( xATHOC.GetUID() != 555555)
xATHOC.SetUser(555555, 'abcdabcd')
}
]>
</functions>
</responsePayload>
</ok>
```

Response with the <noFunctionScript /> node

When the <noFunctionScript /> node is present, a response payload is returned with the following node:

```
<ok>
<systemDate>time of server</systemDate>
<responsePayload>
<userId>user-id</userId>
</responsePayload>
<warnings />
</ok>
```

Error codes

The following user synchronization error codes are defined:

Error code	Message	Description
1300	Missing parameter	No MID
9999	Missing parameter	No client-id

Get user details

This API can be used to obtain basic user details such as the username, display name, etc. See SDK Envelope format specification on page 7.

The payload type is GET-USER. The payload content should be in the following format:

```
<user>
<userId>userId</userId>
<mid>user name</mid>
</user>
```

Mandatory nodes

Either the <mid> or the <userId> node must be populated. If both are populated, <userId> will take priority over <mid>.

- <mid> is a mapping-id used to uniquely identify the user between the systems. The MID value can be any alphanumeric string, between 2 and 80 characters
- <userId> is an existing numeric BlackBerry AtHoc user-identity.

The response payload is returned in the following format:

```
<user>
<userId>userId</userId>
<mid>user name</mid>
<userName>UserLoginName</userName>
<displayName>User Full Name</displayName>
<firstName>User First Name</firstName>
<lastName>User Last Name</lastName>
<pin>User PIN</pin>
<status> VLD | DSB </status>
<customFields>
<field id="numeric id" commonName="">Field value(s)</field>
...
```

```

</customFields>
<devices>
<device id="numeric id" commonName="">Device Address</device>
...
</devices>
</user>

```

User device subscription

This API allows an external application to get the delivery device addresses of a user.

Get user device addresses

This API call obtains the devices and addresses to which the user is subscribed. To learn about the XML format, see [SDK envelope request](#).

The payload type is GET-USER-DEVICE-ADDRESS. The payload content uses the following format:

```

<userDeviceAddresses>
<user>
<mid>mid</mid>
<userId>user-id</userId>
</user>
<deliveryPreferences>
<addressId>optional address id</addressId>
</deliveryPreferences>
</userDeviceAddresses>

```

Mandatory nodes

- **<user>**—Either the **<mid>** or the **<userId>** sub-node must be populated. If both are populated, **<userId>** will take priority over **<mid>**.
- **<mid>**—A mapping ID, to uniquely identify the user between the systems.
- **<userId>**—An existing BlackBerry AtHoc user identity.

Optional nodes

- **<addressId>** (optional)—Gets details for the specified address only. If not supplied, returns details for all the device addresses for the user.

The SDK response will be:

```

<reponsePayload>
<device>
<deviceId>device id</deviceId>
<deviceCommonName>device common name</deviceCommonName>
<deviceName>device name</deviceName>
<isMandatory>Y|N</isMandatory>
<isEditable>Y|N</isEditable>
<deviceAddress>
<addressId>address id</addressId>
<address>user@company.com</address>
<addressLabel>display name for address</addressLabel>
<isPrimary>Y/N</isPrimary>
<isValidated> Y/N </isValidated>
</deviceAddress>
</device>
</device>

```

```
...  
</device>  
...  
</responsePayload>
```

Diagnostic log

This API call provides information about events in the alerting process.

Log diagnostic information

This API allows a remote SDK client to log information useful for application and system diagnostic purposes. See the SDK Envelope format specification in this guide for more information.

The payload type is DIAGNOSTIC-LOG. The payload XML definition is:

```
<diagnosticLog>
  <type>Error</type>
  <server>localhost</server>
  <process>YourProcess</process>
  <assembly>YourAssembly.exe</assembly>
  <module>YourModule.cs</module>
  <member>YourModuleMember</member>
  <message>Your error message.</message>
  <thread>ThreadName</thread>
  <threadId>Thread ID</threadId>
  <stackTrace/>
  <exception/>
  <customContext/>
  <operationStack/>
  <processId/>
</diagnosticLog>
```

Mandatory nodes

Node	Definition
<type>	Diagnostic log type. Possible values for the type field are: • Error • Warning • Information • Verbose
<message>	Diagnostic message the client application wants to log.

Optional nodes

Node	Definition
<server>	Client application's host machine name or IP.
<process>	Client application's process name.
<assembly>	Client application's assembly name.
<module>	Client application's module name.

Node	Definition
<member>	Client application's member method name.
<thread>	Name of the thread running in the client application.
<threadId>	ID of the thread running in the client application. Validated as 32-bit integer if provided, otherwise ignored.
<stackTrace>	Set the stack trace if available.
<exception>	Exception information if an exception occurred.
<customContext>	Client custom context.
<operationStack>	Client operation stack.
<processId>	Client's process ID. Validated as 32-bit integer if provided, otherwise ignored.

Manage API Access and Security

This section provides information about how to validate SDK requests and restrict API access by IP.

Validate SDK requests

There are three ways to provide validation to the server that an SDK request is coming from an operator with valid access permissions:

- Username and password
- Session ID
- Access token

These are all placed inside the validation section of the payload.

Validate with a password and username

You can provide a username and password in the <validation> attribute of any payload to validate an SDK request:

```
<AtHocSdk>
<client>organizationID</client>
<validation>
<username>OperatorUserName</name>
<password>OperatorPassword</password>
</validation>
<payload type="GET-DISTRIBUTION-LISTS">
</payload>
</AtHocSdk>
```

Validate with a session ID

You can request a session to be created on the server through the login API. The login API returns a session ID that can be used instead of a username and password. This enhances security by removing the need to store a username and password in your application.

Login

```
<AtHocSdk>
<client>organizationID</client>
<validation>
<username>OperatorUserName</name>
<password>OperatorPassword</password>
<udid>OperatorDeviceIdentifier</udid>
</validation>
<payload type="Login"/>
</AtHocSdk>
```

This login method returns a response that contains the session ID for the remainder of the operator session.

```
<AtHocSdk>
<client>2060349</client>
<validation>
<sessionId>c345c6f6-a46c-4d41-9bfb-7f1d7b79c528</sessionId>
</validation>
<payload type='LOGOUT' />
```

```
</AtHocSdk>
```

Example payload

Use the session ID returned in the response to call the SDK. The session ID replaces the Username and password in the <validation> attribute.

```
<AtHocSdk><client>organizationID</client>
<validation>
<sessionID>C345c6f6-a46c-4d41-9bfb-7f1d7b79c528</sessionID>
</validation>
<payload type="GET-DISTRIBUTION-LISTS">
</payload>
</AtHocSdk>
```

Logout

After calling the SDK with the session ID, use the logout SDK to invalidate the session ID:

```
<AtHocSdk>
<client>2060349</client>
<validation>
<sessionId>c345c6f6-a46c-4d41-9bfb-7f1d7b79c528</sessionId>
</validation>
<payload type='LOGOUT' />
</AtHocSdk>
```

Validate with an access token

You can use access tokens returned from the Web API V2 authentication service to validate SDK payload requests.

See the "How to authenticate" section of the *BlackBerry AtHoc API Quick Start Guide* for instructions on how to obtain an access token.

Example

```
<AtHocSdk><client>organizationID</client>
<validation>
<accessToken>C345c6f6-a46c-4d41-9bfb-7f1d7b79c528
.
.
.
</accessToken>
</validation>
<payload type="GET-DISTRIBUTION-LISTS">
</payload>
</AtHocSdk>
```

Restrict API access by IP

You must have Administrator permissions to restrict API access by IP.

Posting sources can be restricted by specifying allowed IP addresses in the **Settings > User Authentication > SDK Access Security** section, as shown in the following image:

▼ SDK Access Security

Allowed IP Addresses

List of IP addresses that can call the SDK. Add multiple addresses separated by commas.

Note

1. If no IP addresses are specified, any computer can send API requests subject to username/password restrictions.

2. Each API request must include a username and password to provide secure access to the API and define the rights of specific API requests. You must have SDK User permissions to define the username and password for an API request.

Separate multiple IP addresses with commas. If no IP addresses are specified, it is assumed that posting is not limited by IP. If a POST comes from an IP which is not allowed, the request is ignored.

Sample code

This section gives sample code for interacting with the SDK.

C#

```
using System;
using System.IO;
using System.Net;
using System.Xml;
using System.Web;
using System.Text;

class SDKPost
{
    private const string C_ATHOC_SERVER_LOCATION = "http://athoc.server.com";
    private const string C_EXTENSION_LOCATION = "/sdk/listener/listen.asp";
    private const string C_PROXY_HOST = "";
    private const int C_PROXY_PORT = 80;

    public static void Main()
    {
        //this is our string URL
        String uri = new

        StringBuilder(C_ATHOC_SERVER_LOCATION).Append(C_EXTENSION_LOCATION).ToString();

        //build AtHoc Sdk authentication and user sync request xml
        String postData = new StringBuilder("<AtHocSdk>")
            .Append("<client>2010110</client>")
            .Append("<validation>")
            .Append("<username>SDK Username</username>")
            .Append("<password>SDK Password</password>")
            .Append("</validation>")
            .Append("<payload type='USER-SYNC'>")
            .Append("<userSynchronization>")
            .Append("<mid>johnsmith</mid>")
            .Append("</userSynchronization>")
            .Append("</payload>")
            .Append("</AtHocSdk>").ToString();

        String userProfileXML = PostURL(uri, false, postData);
        System.Console.WriteLine(userProfileXML);
    }

    // PostURL - posts postData to p_PageURL, while considering p_useProxy
    // returns the document from the posted to URL
    public static String PostURL(String p_PageURL, bool p_useProxy, String postData)
    {
        String strXML = "";

        try
        {
            postData = "data=" + HttpUtility.UrlEncode(postData);
            byte[] data = Encoding.ASCII.GetBytes(postData);
```

```

HttpRequest l_WebReq = (HttpRequest)WebRequest.Create(p_PageURL);
l_WebReq.Method = "POST";
l_WebReq.ContentLength = data.Length;
l_WebReq.ContentType = "application/x-www-form-urlencoded";

if (p_useProxy)
{
WebProxy l_proxyObject = new WebProxy(C_PROXY_HOST, C_PROXY_PORT);
l_WebReq.Proxy = l_proxyObject;
}

Stream newStream = l_WebReq.GetRequestStream();
try
{
// Send the data.
newStream.Write(data, 0, data.Length);
}
catch (Exception e)
{
return e.Message;
}
finally
{
newStream.Close();
}

HttpWebResponse l_WebRes = (HttpWebResponse)l_WebReq.GetResponse();
StreamReader l_sr =
new StreamReader(l_WebRes.GetResponseStream(), Encoding.ASCII);
strXML = l_sr.ReadToEnd();
l_sr.Close();
}
catch (Exception err)
{
strXML = "WEB PAGE ERROR : in RetrieveWebPage URL = " +
p_PageURL + " - " + err.ToString();
}
return strXML;
} // of PostURL
}

```

VBScript

```

dim objXMLDoc, strPayload, strSdkUrl, strPid, strSdkUser, strSdkPass

Set objXmlDoc = CreateObject("MSXML2.DOMDocument")
objXmlDoc.async = false

'Config variables
strSdkUrl = "http://athoc.server.com/sdk/listener/listen.asp"
strPid = "2010110"
strSdkUser = "SDK Username"
strSdkPass = "SDK Password"

'Form AtHoc XML payload
strPayload = "<AtHocSdk>" &_

```

```

"<client>" & strPid & "</client>" &_
"<validation>" &_
"<username>" & strSdkUser & "</username>" &_
"<password>" & strSdkPass & "</password>" &_
"</validation>" &_
"<payload type='USER-SYNC'>" &_
"<userSynchronization>" &_
"<mid>joesmith</mid>" &_
"<enabled>N</enabled>" &_
"</userSynchronization>" &_
"</payload>" &_
"</AtHocSdk>"

'Post XML payload to AtHoc listener web page
objXmlDoc.load(strSdkUrl & "?data=" & Escape(strPayload))

'Check for errors and echo status
If objXmlDoc.parseError.errorCode <> 0 Then
WScript.Echo "Error: Could Not Connect to Server: " & Now()
ElseIf Not objXmlDoc.selectSingleNode("//ok") Is Nothing Then
WScript.Echo "API Call Successful: " & Now()
Else
WScript.Echo "ERROR in API call. Error Was: " & objXmlDoc.selectSingleNode("//
error").text & " " & Now()
End If

'Close out variables
Set strPayload = Nothing
Set strSdkUrl = Nothing
Set strPid = Nothing
Set strSdkUser = Nothing
Set strSdkPass = Nothing

```

Internationalization-supported characters

The BlackBerry AtHoc system supports Windows-1252, a set of characters that includes all of the foreign characters that are required for the languages currently supported in the system. For additional details about Windows-1252, see <https://en.wikipedia.org/wiki/Windows-1252>.

In addition, some of the key fields for alerting support the Unicode character set, which is much larger than Windows-1252. This is important because it enables users to create alerts that have delivery locales that are different from the Organization locale. For more details, see the "Selecting a Delivery Locale for an Alert" section of the *BlackBerry AtHoc Localization Guide*.

The following table defines which fields in BlackBerry AtHoc allow which characters:

Text fields or components	Allowed characters	Notes
User Profile fields		
Username	Supported alpha-numeric characters	Avoid space and these characters [] : ; = , + * ? < >
Friendly Name for User Attributes	Supported alpha-numeric characters	Avoid special characters such as `!\$%^()=}{.,; \:?"<> []&. Although the UX supports Unicode, it will break other flows like import, SDK Sync.
Friendly Name for Device	Supported alpha-numeric characters	Avoid special characters such as `!\$%^()=}{.,; \:?"<>
Common Name: Devices, User Attributes, User Attribute Values	Supported alpha-numeric characters	Avoid special characters such as `!\$%^()=}{.,; \:?"<> []&
Values: Device Values, User Attribute Values, Organization Hierarchy	Supported alpha-numeric characters	Keep the values for attributes to supported characters. Special characters create problems when searching for or targeting users. The "&" special character is supported.
Alerting		
Alert Title, Alert Body	Unicode support	—

Text fields or components	Allowed characters	Notes
Response Options	Unicode support	—
SDK payloads		
GET-SCENARIO,GET-SCENARIOS INFOCASTING, GET-HIERARCHY, GET-DISTRIBUTION-LISTS, GET-DEVICES, GET-ALL-CHANNELS	Supported alpha-numeric characters	—

Supported characters

The following alphanumeric characters are supported:

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
 ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖØÙÚÛÜÝÞß àáâãäåæçèéêëìíîïðñóôõöøùúûüýþÿŒœŠšŸŽžf

Phone number validation

Overview

An Emergency Mass Notification System is only as effective as the contact information it contains. For this reason, BlackBerry AtHoc is always looking to improve both the quality and quantity of contact details for end users. BlackBerry AtHoc release 6.1.8.89 added a new phone number validation feature that applies to all phone numbers, no matter which country they belong to. It also enforces clean data wherever data can be entered.

The validation feature gives operators higher confidence before an alert is sent that end users with phone numbers are reachable. One way it does this is by ensuring that end users completing self-service profiles enter actual phone numbers, instead of invalid data such as “No Phone” or “N/A.” Validating phone numbers when they are created in the system makes the alerting process more rapid and efficient by preventing the Telephony Delivery Service from wasting time trying to send telephone notifications to invalid numbers.

BlackBerry AtHoc introduced this feature alongside other improvements for customers operating outside or calling users who are outside the United States. Validated phone numbers can be stored in the internationally recognized E.164 format, ensuring that alerts sent by delivery services deployed throughout the world will reach their destinations.

Areas of the system that validate phone numbers

The following inputs will use the same set of phone number validation rules:

- AtHoc SDK
- LDAP/Active Directory Sync module
- CSV Import
- Self Service
- User Details page in the Management System

Validation rules

The following validation rules are delivered by a third-party open source component. For more information, see: <https://github.com/googlei18n/libphonenumber>.

1. E.164 international format is preferred and is always accepted.
 - The number should start with + followed by the country code and then the full number to call. A maximum of 15 digits can be used.
 - For example: +18884628462
2. Numbers can have an extension.
 - The user interface has a separate field for telephone extensions. When importing numbers, an x should be used to separate the main number from the extension.
 - When dialing, the Telephony Delivery Service will wait for the call to connect before dialing the extension.
 - For example: +18884628462x1340
 - Unlike the phone number field, the extension field is not validated.
3. Numbers not in E.164 are interpreted based on the Default Country Code for the Organization.
 - The Default Country Code can be set on the General Settings screen in the Phone Call Settings section.
 - For example, for the Country Code “US,” the following rules apply:
 - If the number starts with 011, which is the international exit code from within US, it will be replaced with +.
 - If the number contains only 10 digits, it will be stored as +1 followed by the number.
 - If the number contains 11 digits and starts with 1, it will be stored as +1 followed by the number.
 - For example: (888) 462-8462 will be interpreted as +18884628462

4. Common formatting punctuation is ignored.
 - The following characters are removed: ().-_
 - For example: +1 (888) 462-8462 will be interpreted as +18884628462.
 - If you are using control characters such as , (comma) or # (pound sign), they must be in the extension field.
5. If the number contains letters, they will be converted to numbers according to a standard keypad.
 - For example: (888) Go AtHoc will be converted to +18884628462.
6. If the number starts with +, it will be assumed to be an international number.
 - For example: A number starting with +440 will dial the UK, even though 440 is a valid US area code.

Best practices

Send all numbers in E.164 format. Although E.164 format is not required, it is the best way to send a number to the system, especially if user data can contain numbers from different countries.

Make sure you set the correct Default Country Code in the Phone Call Settings section on the General Settings screen. This specifies what country is the default for user-entered phone numbers. This also is used to interpret phone numbers that are not in E.164 format.

If the number contains any special control characters that must be dialed, such as , (comma) ; (semicolon) * (asterisk) or # (pound sign), the characters must be part of the extension. This is especially important for numbers that connect to a conference bridge.

Special note

If you are unable to comply with the new validation rules, fields that do not contain valid phone numbers will not be updated.

For AtHoc IWS versions 6.1.8.88 and earlier , you must continue to use 011 instead of + at the beginning of all international phone numbers.

AtHoc IWS version 6.1.8.89 and beyond fully support the leading + method. Dialing 011 will continue to be supported after upgrade to 6.1.8.89 for organizations with a US country code since 001 is the US exit code.

BlackBerry AtHoc customer portal

BlackBerry AtHoc customers can obtain more information about BlackBerry AtHoc products or get answers to questions about their BlackBerry AtHoc systems through the Customer Portal:

<https://support.athoc.com/customer-support-portal.html>

The BlackBerry AtHoc Customer Portal also provides support via computer-based training, Operator checklists, best practice resources, reference manuals, and users guides.

Legal notices

Copyright © 2019 BlackBerry Limited. All Rights Reserved.

This document may not be copied, disclosed, transferred, or modified without the prior written consent of BlackBerry Limited. While all content is believed to be correct at the time of publication, it is provided as general purpose information. The content is subject to change without notice and is provided “as is” and with no expressed or implied warranties whatsoever, including, but not limited to, a warranty for accuracy made by BlackBerry Limited. The software described in this document is provided under written license only, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. Unauthorized use of software or its documentation can result in civil damages and criminal prosecution.

Trademarks

Trademarks, including but not limited to ATHOC, EMBLEM Design, ATHOC & Design and the PURPLE GLOBE Design are the trademarks or registered trademarks of BlackBerry Limited, its subsidiaries and/or affiliates, used under license, and the exclusive rights to such trademarks are expressly reserved. All other trademarks are the property of their respective owners. Users are not permitted to use these marks without the prior written consent of AtHoc or such third party which may own the mark.

This product includes software developed by Microsoft (<http://www.microsoft.com>).

This product includes software developed by Intel (<http://www.intel.com>).

This product includes software developed by BroadCom (<http://www.broadcom.com>).

All other trademarks mentioned in this document are the property of their respective owners.

Patents

This product includes technology protected under patents and pending patents.

BlackBerry Solution License Agreement

<https://us.blackberry.com/legal/blackberry-solution-license-agreement>

Contact Information

BlackBerry AtHoc

311 Fairchild Drive

Mountain View, CA 94043

Tel: 1-650-685-3000

Email: athocsupport@blackberry.com

Web: <http://www.athoc.com>