www.novell.com/documentation

Stubbing

GroupWise® Developer Kit

November 2012

Novell®

Legal Notices

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell developer products, and to get updates, see the Novell Developer Web site (http://www.novell.com/developer). To access online documentation for Novell products, see the Novell Documentation Web site (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

GroupWise stubbing provides access to events or actions that occur in a GroupWise user's mailbox.

IMPORTANT: Unless otherwise indicated, the features in GroupWise stubbing work with GroupWise 8 and later versions.

This guide contains the following sections:

- Chapter 1, "Overview," on page 7
- Chapter 2, "Methods," on page 17
- Chapter 3, "Schema Elements," on page 27
- Appendix A, "Revision History," on page 97

Audience

This guide is intended for GroupWise developers.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation, or go to Novell Documentation Feedback (http://www.novell.com/documentation/feedback.html) and enter your comments there.

Additional Documentation

For additional GroupWise SDK documentation, see the Novell Developer Web site (http://www.novell.com/developer).

1 Overview

Stubbing facilitates access to an archive server to enhance the flow of information from the archive server to a GroupWise system.

This section contains the following sections:

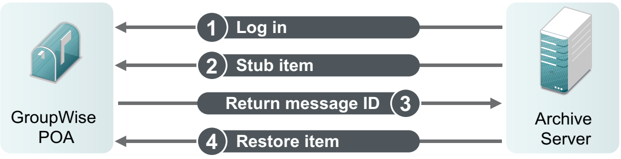
- Section 1.1, "Stubbing Overview," on page 7
- Section 1.2, "Enabling Stubbing on a POA," on page 10
- Section 1.3, "Authentication," on page 10
- Section 1.4, "Message Bodies," on page 14
- Section 1.5, "Session Id," on page 16

1.1 Stubbing Overview

GroupWise stubbing allows a GroupWise Windows client to directly access third-party archive items.

A stub is an archived item, where all the attachments (files, distribution lists, message body) have been removed from the GroupWise store and placed into a third-party store. The archive server communicates with the POA to create a stub.

Figure 1-1 *Creating a Stub Item*

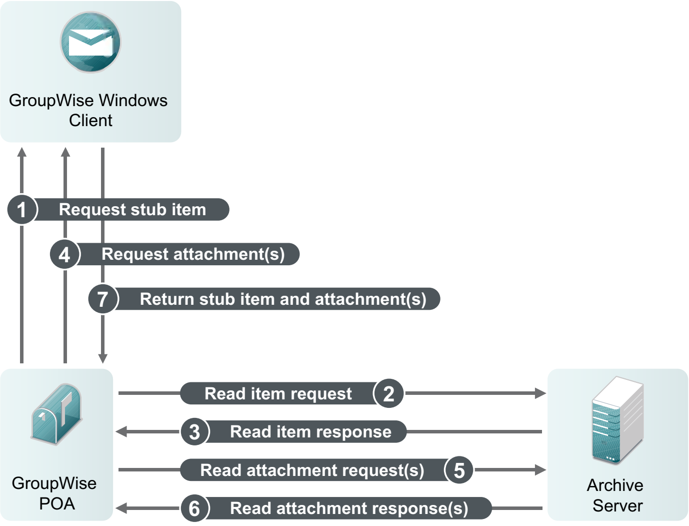


For more information, see Section 1.1.2, "Create," on page 9.

1. The archive server logs in to the POA to create a stub for an archived item.
2. The archive server transfers all item content into its archive store.
3. The POA provides the original message ID as the ID for the archived content.
4. If needed, the archive server can restore the item content by replacing the stub with the original item.

When a GroupWise client user requests the information from a stubbed item, the POA sends a request to the archive server requesting the stubbed information.

Figure 1-2 *Example of a Stub Item Request*



1. When a GroupWise user wants to view an item that has been stubbed, the GroupWise client requests the item from the POA.
2. The POA requests the item from the archive server.
3. The archive server returns initial item content including the distribution list.
4. When the GroupWise client receives the initial item content, it requests item attachments.

   The message body text is an attachment, as well as any files that the item sender attached.
5. The POA requests the attachments from the archive server.
6. The archive server returns the attachments (item message body and pointers to any files that the sender attached).

   If any attachment is larger than 32K, multiple requests and responses are required to deliver the attachment.

   The content of attached files is not returned unless the GroupWise user views the attachment.
7. The POA delivers the item to the GroupWise client for viewing.

For more information, see Section 1.1.3, “Read,” on page 9.

The administrator must enable stubbing in ConsoleOne before it can be used. For more information see Section 1.2, “Enabling Stubbing on a POA,” on page 10.

- Section 1.1.1, “Login,” on page 9
- Section 1.1.2, “Create,” on page 9
- Section 1.1.3, “Read,” on page 9
- Section 1.1.4, “Find,” on page 9
- Section 1.1.5, “Logout,” on page 9

1.1.1 Login

Prior to the POA sending any requests to the archive server, a session must be established with the archive server by sending a `loginRequest` SOAP command to the archive server. This establishes a session (denoted by a session key) that is used by the remainder of the commands.

1.1.2 Create

Stubs are created by the archive server through the `stubItemRequest` SOAP interface call. A stub is not the same item as the original item. The original item is deleted and replaced by the stub. The stub retains the original GroupWise message ID that it had before it was stubbed. A stubbed item has a flag denoting a stubbed item. Each stubbed item is identified with a unique icon in the GroupWise client.

Only the GroupWise Windows client supports opening stubbed items. They cannot be opened in GroupWise WebAccess.

1.1.3 Read

When a user clicks an item in the Windows client to open the item, the Windows client sends a request to the POA to read the item. The POA sends a `readItemRequest` SOAP request to the archive server to return the original item data in a `readItemResponse` SOAP response.

The original item data is returned with the exception of the attachment data. The SOAP response returns information regarding the attachments, but the Windows client must to perform additional `readAttachmentRequests` SOAP requests to retrieve the attachment data. The archive server returns this data in `readAttachmentResponse` SOAP responses. The Windows client asks for chunks of data until all of the attachment data has been returned.

1.1.4 Find

The Windows client allows the user to search the archive server. The archive server is displayed as another source in the Find window.

During the find, the POA sends `readArchiveRequest` SOAP requests to the archive server. The POA puts the `readArchiveResponse` SOAP response data into the find results displayed by the client. The POA continues to request more data until the archive server indicates there is no more data. If a user cancels the find, a cancel request is transmitted to the POA from the Windows client. The POA then sends a `cancelReadRequest` SOAP request to the archive server.

The archive server can return an error code in any SOAP response, which cancels the operation, and in most cases is displayed by the client.

1.1.5 Logout

When the user logs out of the Windows client, the POA sends SOAP `logoutRequest` messages to the archive server to close all active sessions.

1.2 Enabling Stubbing on a POA

Stubbing requires the use of a trusted application. This section assumes you have already set up and properly configured a trusted application. For help creating a trusted application see, "Trusted Applications (http://www.novell.com/documentation/gw8/gw8_admin/data/ake1tw1.html)" in the *GroupWise 8 Administration Guide (http://www.novell.com/documentation/gw8/gw8_admin/data/a2zvyc4.html)*.

- GroupWise 2012: "Trusted Applications" in "System" in the *GroupWise 2012 Administration Guide*
- GroupWise 8: "Trusted Applications" in "System" in the *GroupWise 8 Administration Guide*

To enable stubbing:

1. In ConsoleOne, click *GroupWise System* in the GroupWise View.
2. Click *Tools* > *GroupWise System Operations* > *Trusted Applications*.
3. Select the trusted application you want to use, then click *Edit*.
4. Select *Provides Message Retention Service*.
5. Select *Allow access to Archive Service*, then specify the IP address of the archive server.
6. Add `/gwarchive` to the end of the IP address.

   For example, if the IP address of your archive server is `archiveserver.com:7151`, you specify the archive server address as follows:

   ```
   http://www.archiveserver.com:7151/gwarchive
   ```

7. Click *OK* to save the changes.

After stubbing is enabled, you must select the correct trusted application for each GroupWise system, domain, or post office you are using with stubbing.

1. In ConsoleOne, select the GroupWise system, domain, or post office.
2. Click *Tools* > *GroupWise System Operations* > *System Preferences*.
3. Click *Archive Service Settings*, then select the properly configured trusted application in the *Archive Service Trusted Application* drop-down menu.
4. Click *OK* to save your changes.

1.3 Authentication

Authentication occurs through a series of SOAP requests. The following sections contain sample authentications.

- Section 1.3.1, "Login Request from GroupWise," on page 11
- Section 1.3.2, "Read Item Request from GroupWise," on page 11
- Section 1.3.3, "Read Attachment Request," on page 12
- Section 1.3.4, "Logout Request," on page 13

1.3.1 Login Request from GroupWise

Login Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gwar="http://
schemas.novell.com/2007/05/GroupWise/archive">
<env:Body>
<gwar:loginRequest>
<gwar:auth xsi:type="gwar:TrustedApplication">
<gwar:username>user1.po.primary.6A6A0AB0-0200-0000-B188-29BCB65623D7</
gwar:username>
<gwar:name>Retain</gwar:name>
<gwar:key>C57703410525000090EAAD001600F300C57703420525000090EAAD001600F300</
gwar:key>
</gwar:auth>
<gwar:language>en</gwar:language>
<gwar:version>1.0</gwar:version>
</gwar:loginRequest>
</env:Body>
</env:Envelope>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gwar="http://
schemas.novell.com/2007/05/GroupWise/archive">
<env:Body>
<gwar:loginRequest>
<gwar:auth xsi:type="gwar:TrustedApplication">
<gwar:username>user1.po.primary.6A6A0AB0-0200-0000-B188-29BCB65623D7</
gwar:username>
<gwar:name>Retain</gwar:name>
<gwar:key>C57703410525000090EAAD001600F300C57703420525000090EAAD001600F300</
gwar:key>
</gwar:auth>
<gwar:language>en</gwar:language>
<gwar:version>1.0</gwar:version>
</gwar:loginRequest>
</env:Body>
</env:Envelope>
```

1.3.2 Read Item Request from GroupWise

Read Item Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:gwar="http://schemas.novell.com/2007/05/GroupWise/archive">
<env:Header>
<gwar:session>4m532tcksq6ip14dr74sjm-1g18vm</gwar:session>
</env:Header>
<env:Body>
<gwar:readItemRequest>
<gwar:id>4A77FDD5.primary.po.100.1726E36.1.52D.1.52D.1@1</gwar:id>
<gwar:archiveId>rO0ABXQAHmVtYWlsSUQ9NTY5JnN2PTEmdmVuZG9yPXJldGFpbg==</
gwar:archiveId>
<gwar:originalId>4A647C96.$$$.$$$.100.1242424.1.20B.1@1</gwar:originalID>
</gwar:readItemRequest>
</env:Body>
</env:Envelope>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:gwar="http://schemas.novell.com/2007/05/GroupWise/archive">
<env:Header>
<gwar:session>4m532tcksq6ip14dr74sjm-1g18vm</gwar:session>
</env:Header>
<env:Body>
<gwar:readItemRequest>
<gwar:id>4A77FDD5.primary.po.100.1726E36.1.52D.1.52D.1@1</gwar:id>
<gwar:archiveId>rO0ABXQAHmVtYWlsSUQ9NTY5JnN2PTEmdmVuZG9yPXJldGFpbg==</
gwar:archiveId>
<gwar:originalId>4A647C96.$$$.$$$.100.1242424.1.20B.1@1</gwar:originalID>
</gwar:readItemRequest>
</env:Body>
</env:Envelope>
```

1.3.3 Read Attachment Request

- "Read Attachment Request 1" on page 12
- "Read Attachment Request 2" on page 13

Read Attachment Request 1

Read Attachment Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:gwar="http://schemas.novell.com/2007/05/GroupWise/archive">
<env:Header>
<gwar:session>4mknf7tbl9rie14dr74ts7-1g18vm</gwar:session>
</env:Header>
<env:Body>
<gwar:readAttachmentRequest>
<gwar:id>rO0ABXQAb3NlcnZsZXQvcmFyP2RvY0lEPTU1M0QwRTA1M0ZFMEFGMUI1Qzk4ODYzMDVGRDM0Q
zQ2QzVFMTIyRTZEQzM1Njg5MTkyOUJEQUUzNzEyRkU3NkImdGFnSUQ9JmRvY1NpemU9MTA1NTQyJmd6aXA
9MQ==</gwar:di>
<gwar:offset>0</gwar:offset>
<gwar:length>32767</gwar:length>
</gwar:readAttachmentRequest>
</env:Body>
</env:Envelope>
```

Response:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body><readAttachmentResponse xmlns="http://schemas.novell.com/2007/05/
GroupWise/archive">
<part length="32767" offset="32767">
/9j/4AAQSkZJRgABAgEAYABgAAD/7RBKUGhvdG9zaG9wIDMuMAA4QklNA+0KUmVzb2x1dGlv...
</part>
<status>
<code>0</code>
</status>
</readAttachmentRequest>
</S:Body>
</S:Envelope>
```

Read Attachment Request 2

Read Attachment Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:gwar="http://schemas.novell.com/2007/05/GroupWise/archive">
<env:Header>
<gwar:session>4mknf7tbl9rie14dr74ts7-1g18vm</gwar:session>
</env:Header>
<env:Body>
<gwar:readAttachmentRequest>
<gwar:id>rO0ABXQAb3NlcnZsZXQvcmFyP2RvY0lEPTU1M0QwRTA1M0ZFMEFGMUI1Qzk4ODYzMDVGRDM0Q
zQ2QzVFMTIyRTZEQzM1Njg5MTkyOUJEQUUzNzEyRkU3NkImdGFnSUQ9JmRvY1NpemU9MTA1NTQyJmd6aXA
9MQ==</gwar:di>
<gwar:offset>32767</gwar:offset>
<gwar:length>32767</gwar:length>
</gwar:readAttachmentRequest>
</env:Body>
</env:Envelope>
```

Response:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body><readAttachmentResponse xmlns="http://schemas.novell.com/2007/05/
GroupWise/archive">
<part length="32767" offset="65534">
ztCmtFb4dJ2gacp6tLIaZBP0wJHH8pEdjAtAkg9iNO...
</part>
<status>
<code>0</code>
</status>
</readAttachmentRequest>
</S:Body>
</S:Envelope>
```

1.3.4 Logout Request

Logout Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gwar="http://
schemas.novell.com/2007/05/GroupWise/archive">
<env:Header>
<gwar:session>4mknf7tbl9rie14dr74ts71g18vm</gwar:session>
</env:Header>
<env:Body>
<gwar:logoutRequest></gwar:logoutRequest>
</env:Body>
</env:Envelope>
```

Response:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
"><S:Body><logoutResponse xmlns="http://schemas.novell.com/2007/05/GroupWise/
archive">
<status>
<code>0</code>
</status>
</logoutRequest>
</S:Body>
</S:Envelope>
```

1.4 Message Bodies

Message bodies can be in the following formats:

- Plain text (no formatting is embedded in the file)
- RTF (rich text formatting)
- HTML

The following sections describe how to deal with large message bodies and HTML message bodies.

- Section 1.4.1, "Large Message Bodies," on page 14
- Section 1.4.2, "HTML Message Bodies," on page 15

1.4.1 Large Message Bodies

To retrieve message bodies, the message keyword should be included in the view. For message bodies smaller than 32 KB, the message body is returned in the message element itself. For message bodies greater than 32 KB, a message ID attribute is returned instead of the message body. If a message ID is returned, call "getAttachmentRequest" to download the message body.

The following example returns a large message body. The ID is an attribute on the message element, and the length of the file is just over 1 MB.

```
<items>
   <item type="Mail">
      <id>452A4D93.domain1.po1.100.1647535.1.635.1@1:7.
                     domain1.po1.100.0.1.0.1@16</id>
      ...
      <subject>Large Message Body</subject>
      <message>
         <part contentType="text/plain" length="1008596" id="452A4D93.
         domain1.po1.200.20000B2.1.374.1@65:452A4D93.domain1.po1.100.
           1647535.1.635.1@1:7.domain1.po1.100.0.1.0.1@16" />
      </message>
      ...
   </item>
</items>
```

To read the message body, call "getAttachmentRequest" with the ID of the message body. GetAttachmentRequest allows you to read the message body in chunks by providing an offset and a length. The offset is the location in the file to start reading. The length is the number of bytes to read. In the first call to getAttachmentRequest, the offset is 0 and the length is 16K.

```
<getAttachmentRequest>
   <id>452A4D93.domain1.po1.200.20000B2.1.374.1@65:452A4D93.domain1.
       po1.100.1647535.1.635.1@1:7.domain1.po1.100.0.1.0.1@16</id>
   <offset>0</offset>
   <length>16384</length>
</getAttachmentRequest>
```

In the following example, the offset is an attribute on the part element. In the next call to getAttachmentRequest, pass in the returned offset value as the offset. The reason the offset can be different than the length passed in is because message bodies are stored in GroupWise in an internal format. After you read in the requested amount of data from the GroupWise store, the data is translated and the size changes.

```
<getAttachmentResponse >
   <part length="16180" offset="16379">
        Njk3MzcwNkM2MTc5MjA2NDZGNjM3NTZENjU2RTc0NzMyMD
        c5NkY3NTIwNjg2MTc2NjUyMDY
        ...
        EAgRk9MREVSX1JFQ09SRCA=</part>
      <gwm:status>
   <code>0</code>
   </status>
</getAttachmentResponse>
```

The next time getAttachmentRequest is called, it looks like the following example. The offset on the call matches the offset returned in the last getAttachmentResponse.

```
<getAttachmentRequest>
   <id>452A4D93.domain1.po1.200.20000B2.1.374.1@65:452A4D93.
   domain1.po1.100.1647535.1.635.1@1:7.domain1.po1.100.0.1.0.1@16</id>
   <offset>16379</offset>
   <length>16384</length>
</getAttachmentRequest>
```

The example above is only for message bodies. Normal attachments return the same offset as the specified length.

1.4.2 HTML Message Bodies

Message bodies can be in HTML format. HTML messages are returned as the first attachment and are named `text.htm`. Any embedded images or associated files are also downloaded as attachments. These HTML associated files are all marked hidden.

The HTML message body and accompanying files are grouped together. The `text.htm` file is first, followed by the other associated files. Each of the accompanying attachments has a contentId element. Loop through the attachments until you come to an attachment that does not have a contentId element or until you come to the end of the attachments. If you want to create an HTML message body in SOAP, you need to add the attachments in the same order and add the contentId element for the accompanying files.

In the following example, an item has an HTML message body with an embedded image. The attachments section contains a `text.htm` file. Attachment 2 is an embedded image in the HTML attachment. Also notice that the message element itself has a message body. This is the attachment in plain text format.

```
<item type="Mail">
   <id>452A6054.domain1.po1.100.1647535.1.644.1@1:7.
           domain1.po1.100.0.1.0.1@16</id>
   ...
   <message>
      <part contentType="text/plain"
       length="284">e1xydGYxXGRlZmYwe1xmb250dGJse1xmMFxmbmlsXGZj
       aGFyc2V0MSBUYWhvbWE7fX17XGNvbG9ydGJsO1xyZWQyNTVcZ3JlZW4wX
       GJsdWUwO1xyZWQyNTVcZ3JlZW4wXGJsdWUwO317XHN0eWxlc2hlZXR7XGZz
       MTYgR3JvdXBXaXNlVmlldzt9fVxmczE2IEhUTUwgbWVzc2FnZSB3aXRoIGF
       uIGltYWdlLlxwYXIgIFxwYXIgIFxwYXIgQWZ0ZXIgdGhlIHBpY3R1
       cmUuXHBhciB9</part>
   </message>
   <attachments>
      <attachment>
         <id>452A6054.domain1.po1.200.20000B2.1.398.1@45:452A6054.
                domain1.po1.100.1647535.1.644.1@1:7.domain1.po1.
                100.0.1.0.1@16</id>
         <name>Text.htm</name>
         <contentType>TEXT/HTML</contentType>
         <size>409</size>
         <date>2012-10-09T14:44:35Z</date>
         <hidden>1</hidden>
      </attachment>
      <attachment>
         <id>452A6054.domain1.po1.200.20000B2.1.399.1@45:452A6054.
               domain1.po1.100.1647535.1.644.1@1:7.domain1.po1.
               100.0.1.0.1@16</id>
         <name>IMAGE.gif</name>
         <contentId><UQSTEWRZPCCH.appt.gif></contentId>
         <contentType>IMAGE/gif</contentType>
         <size>143</size>
         <date>2012-10-09T14:44:35Z</date>
         <hidden>1</hidden>
      </attachment>
   </attachments>
   ...
</item>
```

To retrieve the HTML message, call "getAttachmentRequest" with the ID of the `text.htm` file. Your method should look similar to the following:

```
<getAttachmentRequest>
   <id>452A6054.domain1.po1.200.20000B2.1.398.1@45:452A6054.
         domain1.po1.100.1647535.1.644.1@1:7.domain1.po1.
         100.0.1.0.1@16</id>
</getAttachmentRequest>

<getAttachmentResponse>
   <part length="548"
      offset="409">PEhUTUw+PEhFQUQ+DQo8TUVUQSBodHRwLWVxdWl2PUN
      vbnRlbnQtVHlwZSB
      ...
      ERJVj5BZnRlciB0aGUgcGljdHVyZS48L0RJVj48L0JPRFk
     +PC9IVE1MPg==
   </part>
   <status>
      <code>0</code>
   </status>
</getAttachmentResponse>
```

The actual HTML message follows. The embedded image below has a src value equal to the contentId of the second attachment in the preceding example.

```
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<META content="MSHTML 6.00.2900.2963" name=GENERATOR></HEAD>
<BODY style="MARGIN: 4px 4px 1px; FONT: 10pt Tahoma">
<DIV>HTML message with an image.</DIV>
<DIV>&nbsp;</DIV>
<DIV><IMG alt="" hspace=0 src="cid:UQSTEWRZPCCH.appt.gif" align=baseline
border=0></DIV>
<DIV>&nbsp;</DIV>
```

1.5 Session Id

The session ID is returned by the archive server in a `loginResponse` when the POA logs into the server using a `loginRequest`. The session ID uniquely identifies the session and is passed in all subsequent SOAP calls to the archive server. The POA is multi-threaded, so it is possible to have several sessions and session IDs active with the POA at any time. There can also be several sessions active with the same user ID at the same time.

2 Methods

This section contains the following GroupWise stubbing methods, which are defined in the `archive.xsd` schema file:

- "cancelReadRequest" on page 18
- "loginRequest" on page 19
- "logoutRequest" on page 20
- "readArchiveRequest" on page 21
- "readAttachmentRequest" on page 22
- "readItemRequest" on page 23
- "restoreRequest" on page 24
- "stubRequest" on page 25

cancelReadRequest

Cancels a read request.

Request

```
<cancelReadRequest>
   <cursor type="int"/>
</cancelReadRequest>
```

Response

```
<cancelReadResponse>
   <status type="Status"/>
</cancelReadResponse>
```

Elements

cursor

Specifies the identifier for the archive server cursor resource that is to be canceled.

status

Returns the success or failure of the method.

loginRequest

Used to authenticate to the archive server. Only TrustedApplication login is supported to the archive server. The user name has the user name, post office, and domain separated by dots. For example, username.postoffice1.domain.

Request

```
<loginRequest>
   <auth type="Authentication"/>
   <language type="language"/>
   <version type="decimal"/>
</loginRequest>
```

Response

```
<loginResponse>
   <session type="string"/>
</loginResponse>
```

Elements

auth

Specifies the type of authentication as TrustedApplication.

language

The language of the user or the language of the POA if the user language is not specified.

version

The version passed in the request. The current version passed is 1.0.

session

Specifies and links the user with the login instance on the archive server.

logoutRequest

Logs out the current user.

Request

```
<logoutRequest>
   <session type="string"/>
</logoutRequest>
```

Response

```
<logoutResponse>
   <status type="Status"/>
</logoutResponse>
```

Elements

session

Specifies and links the user with the logout instance on the archive server.

status

Returns the success or failure of the method.

readArchiveRequest

Reads from the archive for the current cursor position.

Request

```
<readArchiveRequest>
   <filter type="Filter"/>
   <cursor type="int"/>
   <count type="int"/>
</readArchiveRequest>
```

Response

```
<readArchiveResponse>
   <cursor type="int"/>
   <items type="ItemList"/>
   <status type="Status"/>
</readArchiveResponse>
```

Elements

count

Specifies the number of items to be read. The cursor is then repositioned by the number in the count element.

cursor

On the first call, the cursor is 0. This indicates a new read (cursor) on the archive server. The archive server returns a value for the cursor in the ReadArchiveResponse. That value is passed in the next readArchiveRequest call. The POA continues to call readArchiveRequest until the archive server passes back 0 for the cursor element in readArchiveResponse. That signals there is no more data.

filter

Specifies the filter.

status

Returns the success or failure of the method.

Remarks

The readArchiveRequest works like the cursor calls in the standard SOAP documentation, but the readArchiveRequest handles the createCursorRequest and readCursorRequest functionality in one call. The archive usually signals the end of data by passing 0 back in the cursor element of the readArchiveResponse.

Because of this, there is no need to do a destroyCursorRequest call as in the standard SOAP API. It is possible for the POA to cancel the reading of data. To do so, the POA passes the cursor value in the cancelReadRequest. This signals the archive server that the read is done and the archive server can free any resources involved in the read.

readAttachmentRequest

Returns the selected file attachment.

Request

```
<readAttachmentRequest>
   <id type="uid"/>
   <offset type="int"/>
   <length type="int"/>
</readAttachmentRequest>
```

Response

```
<readAttachmentResponse>
   <status type="Status"/>
</readAttachmentResponse>
```

Elements

id

Specifies the ID of the attachment to read.

length

Specifies the number of bytes to read from the offset.

offset

Specifies the starting position for reading the attachment.

status

Returns the success or failure of the method.

readItemRequest

Returns the item specified by the ID.

Request

```
<readItemRequest>
   <id type="uid"/>
   <archiveId type="string"/>
   <origionalId type="string"/>
</readItemRequest>
```

Response

```
<readItemResponse>
   <item type="Item"/>
   <status type="Status"/>
</readItemResponse>
```

Elements

archivedId

When an item is stubbed, the archive server can assign an ID that it understands. GroupWise does not know the format of the string. If the string is available, the POA passes it to the archive server.

id

The GroupWise ID of the stubbed item.

item

Returns the specified item.

originalId

The ID of the original GroupWise item before it was stubbed. If it is available, the POA passes it to the archive server.

status

Returns the success or failure of the method.

restoreRequest

The POA informs the archive server to restore the original item, with its attachments. This call is not yet implemented.

Request

```
<restoreRequest>
   <id type="uid"/>
   <stub type="boolean"/>
</restoreRequest>
```

Response

```
<restoreResponse>
   <status type="Status"/>
</restoreResponse>
```

Elements

id

The GroupWise ID of the stubbed item.

status

Returns the success or failure of the method.

stubRequest

The POA asks the archive server to stub the selected item. This call is not yet implemented.

Request

```
<stubRequest>
   <id type="uid"/>
</stubRequest>
```

Response

```
<stubResponse>
   <status type="Status"/>
</stubResponse>
```

Elements

id

Specifies the original item ID.

status

Returns the success or failure of the method.

3 Schema Elements

This section defines the elements and attributes that are defined in the GroupWise schemas.

- Section 3.1, "Simple Objects," on page 27
- Section 3.2, "Complex Objects," on page 28

3.1 Simple Objects

The following simple objects can be referenced from the complex objects that are listed in Section 3.2, "Complex Objects," on page 28:

- "acceptLevel" on page 28

acceptLevel

Specifies how an appointment shows in a busy search.

Syntax

```
<acceptLevel type="AcceptLevel"/>
```

3.2 Complex Objects

The following complex objects can reference any of the simple objects listed in Section 3.1, "Simple Objects," on page 27:

- "Alarm" on page 30
- "Appointment" on page 31
- "AttachmentID" on page 32
- "AttachmentInfo" on page 33
- "AttachmentItemInfo" on page 34
- "Authentication" on page 35
- "BoxEntry" on page 36
- "CalendarItem" on page 38
- "CategoryRefList" on page 39
- "ContainerItem" on page 40
- "ContainerRef" on page 41
- "Custom" on page 42
- "CustomList" on page 43
- "CustomType" on page 44
- "Day" on page 45
- "DayOfWeek" on page 46
- "Distribution" on page 47
- "DistributionType" on page 48
- "DocumentRef" on page 49
- "Filter" on page 50
- "FilterDate" on page 51
- "FilterElement" on page 52
- "FilterEntry" on page 53
- "FilterGroup" on page 54
- "FilterOp" on page 55
- "From" on page 62
- "GMTOffset" on page 63
- "Hour" on page 64
- "Item" on page 65

Alarm

Describes the alarm on an appointment.

Syntax

```
<Alarm>
<extension base="int">
<enabled type="boolean"/>
```

Definitions

value

Specifies the number of seconds before the appointment.

enabled

Specifies if the alarm is enabled.

Appointment

Describes a GroupWise appointment.

Syntax

```
<Appointment>
<extension base="CalendarItem">
<ref="startDate"/>
<ref="endDate"/>
<startDay type="date"/>
<endDay type="date"/>
<ref="acceptLevel"/>
<alarm type="Alarm"/>
<allDayEvent type="boolean"/>
<place type="string"/>
<timezone type="Timezone"/>
```

Definitions

CalendarItem

Specifies the CalendarItem (page 38) element. Appointment extends CalendarItem.

startDate

Specifies the start of an appointment.

endDate

Specifies the end of the appointment.

startDay

Specifies the startDate of an appointment (without a time value).

endDay

Specifies the endDate of an appointment (without a time value).

acceptLevel

Specifies how the appointment appears in a busy search.

alarm

Specifies the alarm for the appointment.

allDayEvent

Specifies if the appointment is an all-day event.

place

Specifies where the appointment is to take place.

timezone

Specifies the time zone for the person who scheduled the appointment.

AttachmentID

Uniquely identifies an attachment.

Syntax

```
<AttachmentID>
<extension base="uid">
<attribute name="itemReference" type="boolean"/>
```

Definitions

uid

Specifies the unique identifier for the attachment.

itemReference

Specifies that the attachment is an embedded item (such as, mail, appointment, task, or note). If the item is an itemReference, call “getItemRequest” to retrieve the item. If the item is an attachment, call “getAttachmentRequest” to retrieve the item.

AttachmentInfo

Defines a list of attachments.

Syntax

```
<AttachmentInfo>
<attachment type="AttachmentItemInfo" maxOccurs="unbounded"/>
<isPersonal type="boolean"/>
<hash type="string"/>
```

Definitions

isPersonal

Users can add personal attachments to receive items. This value is true if the attachment is a personal attachment.

hash

A 16-byte hex hash value.

AttachmentItemInfo

Describes an attachment.

Syntax

```
<AttachmentItemInfo>
<id type="AttachmentID"/>
<ref="name"/>
<contentId type="string"/>
<contentType type="string"/>
<size type="unsignedInt"/>
<date type="dateTime"/>
<data type="base64Binary"/>
<hidden type="boolean"/>
```

Definitions

id

Specifies the attachment.

name

Specifies the name of the attachment (usually the file name).

contentId

Specifies the MIME content ID (usually only on the HTML message's related parts).

contentType

Specifies the MIME content type (usually only on the HTML message's related parts).

size

Specifies the size of the attachment. This size value is the native size of the attachment before the Base64 encoding.

date

Specifies the date of the attached file.

data

Specifies the attachment data in Base64 format.

hidden

Specifies if the attachment should be visible to the user. For example, the Windows client hides the `text.htm` file, which is the HTML version of the message body.

Authentication

Contains the base object for plain text, proxy, and trusted application login.

Syntax

```
<username type="string"/>
```

Definitions

userName

The user name used during authentication to GroupWise.

BoxEntry

An abstraction layer that is used to contain information common to all items that can be in a GroupWise mailbox, except address books or address book items.

Syntax

```
<BoxEntry>
<extension base="ContainerItem">
<status type="ItemStatus"/>
<thread type="string"/>
<msgId type="string"/>
<messageId type="string"/>
<clientMessageId type="string"/>
<source type="ItemSource"/>
<returnSentItemsId type="boolean"/>
<delivered type="dateTime"/>
<class type="ItemClass"/>
<security type="ItemSecurity"/>
<comment type="string"/>
```

Definitions

status

Specifies the status of the item: opened, accepted, etc.

thread

Specifies the threading of an item.

msgid

Specifies a shared item in the GroupWise databases. If a message is sent to three people, all three messages have the same msgid.

messageId

Specifies the ID that was returned when this item was retrieved with the IMAP protocol. The messageId is similar to msgid except that the format is different. messageId can help applications match items retrieved with the IMAP and SOAP protocols.

clientMessageId

The same ID that you see when you open an item in the GroupWise Windows client. Select *Properties* and find the Message Id.

source

Specifies the source of the item: received, sent from you, a draft item, or a personal item.

returnSentItemsId

Returns the sent item's ID. This is the ID of the item that is created in the sent items (or the outbox, itemSource=sent) folder. The returnSentItemsId is the first ID returned in the list.

delivered

Specifies when the item was delivered to the mailbox.

class

Specifies the iCalendar (RFC2445) Class.

security

Specifies the privacy level that the sender wants applied to the item. For example, ForYourEyesOnly is a security option.

The security of an item is not enforced by GroupWise. The sender of the item provides the security level they want applied to the item. It is the responsibility of the recipients to abide by the sender's request.

comment

Specifies the comment on a item.

CalendarItem

An abstraction layer that contains common calendar attributes (even if the item is an appointment, note, or task). Rdate, rrule, and exdate elements in CalendarItem specify recurrence.

Syntax

```
<CalendarItem>
<extension base="Mail">
<rdate type="RecurrenceDateType"/>
<rrule type="RecurrenceRule"/>
<exdate type="RecurrenceDateType"/>
<ref="recurrenceKey"/>
<iCalId type="string"/>
```

Definitions

Mail

CalendarItem extends Mail (page 74).

rdate

Specifies a list of recurring dates.

rrule

Specifies a recurrence rule.

exdate

Specifies a list of dates to exclude. For example, if you want to create many appointments based on the rrule element, the exdate can exclude some of the instances in the rrule.

recurrenceKey

Specifies a common key that is shared by all of the items that originated from a recurring calendar item. This key can be used to perform an action, such as accepting recurring items.

iCalId

Specifies the iCalendar (RFC2445) UID.

CategoryRefList

Contains a list of category references.

Syntax

```
<CategoryRefList>
<category type="uid" minOccurs="0" maxOccurs="unbounded"/>
<attribute name="primary" type="uid"/>
```

Definitions

category

Specifies a category.

primary

Specifies the primary category.

ContainerItem

Describes items in a container. A container is an abstraction or a base object. A container can be a folder, address book, document version, or rule.

Syntax

```
<ContainerItem>
<extension base="Item">
<container type="ContainerRef" minOccurs="1" maxOccurs="unbounded"/>
<categories type="CategoryRefList"/>
<created type="dateTime"/>
<customs type="CustomList"/>
<contacts type="ContactRefList"/>
```

Definitions

container

Specifies the container of the item. The item can be in more than one container.

categories

Specifies a list of categories associated with the item.

created

Specifies the date and time that the item was created.

customs

Specifies the list of custom fields associated with the item.

contacts

List of contacts added to a received item.

ContainerRef

Identifies a container.

Syntax

```
<ContainerRef>
<attribute name="deleted" type="dateTime"/>
```

Definitions

deleted

Specifies the time the item was moved to the Trash folder.

Custom

Describes a GroupWise custom field. Currently, only strings are supported.

Syntax

```
<Custom>
<field type="string"/>
<value type="string"/>
<notify type="ReturnNotificationOptions"/>
<locked type="boolean"/>
<adminOnly type="boolean"/>
```

Definitions

field

Specifies the name of the field.

value

Specifies the value of the string field.

notify

Specifies the returnNotification settings for global user settings. This setting corresponds to *Tools* > *Options* > *Send* > *Mail* > *Return Notification*. It also applies to Appointments, Tasks, and Reminder Notes.

locked

Specifies if the value cannot be changed.

adminOnly

Returns true if a setting can only be set by the administrator. The user cannot override this setting.

CustomList

Contains a list of custom fields.

Syntax

```
<CustomList>
<custom type="Custom" minOccurs="0" maxOccurs="unbounded"/>
```

CustomType

Contains an enumeration of customTypes.

Syntax

```
<CustomType>
<enumeration value="String"/>
<enumeration value="Numeric"/>
<enumeration value="Date"/>
<enumeration value="Binary"/>
```

Day

Restricts a day to be between 0 and 30 (1-31).

Syntax

```
<Day>
<restriction base="unsignedByte">
<maxInclusive value="30"/>
```

DayOfWeek

Defines a start or end day and the week in the month. For example, <dayOfWeek occurrence="First">Sunday</dayOfWeek> indicates that an occurrence occurs on Sunday during the first week of the month.

Syntax

```
<DayOfWeek>
<extension base="WeekDay">
<attribute name="occurrence" type="OccurrenceType"/>
```

Definitions

WeekDay

Specifies the days of the week: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday.

occurrence

Specifies the First, Second, Third, Fourth, Fifth, and last week of the month.

Distribution

Contains the distribution and send options for a GroupWise item.

Syntax

```
<Distribution>
<from type="From"/>
<to type="string"/>
<cc type="string"/>
<bc type="string"/>
<recipients type="RecipientList"/>
<sendoptions type="SendOptions"/>
```

Definitions

from

Specifies who sent the item.

to

Specifies the displayed string of the To recipients.

cc

Specifies the displayed string of the CC recipients.

bc

Specifies the displayed string of the BC recipients.

recipients

Specifies the collection of recipients.

sendOptions

Specifies various tracking options.

DistributionType

Specifies the container of the email address (or the user's email address in the TO, CC, or BC fields).

Syntax

```
<DistributionType>
<enumeration value="TO"/>
<enumeration value="CC"/>
<enumeration value="BC"/>
<enumeration value="replyTo"/>
```

Definitions

replyTo

Specifies the replyTo address. The replyTo address is used on a reply by the message.

DocumentRef

Contains a document reference.

Syntax

```
<DocumentRef>
<extension base="Mail">
<library type="NameAndEmail"/>
<documentNumber type="unsignedInt"/>
<filename type="string"/>
<documentTypeName type="string"/>
<author type="NameAndEmail"/>
<creator type="NameAndEmail"/>
<officialVersion type="unsignedInt"/>
<currentVersion type="unsignedInt"/>
<versionNumber type="unsignedInt"/>
<versionDescription/>
<fileSize type="unsignedInt"/>
<acl type="AccessControlList"/>
```

Definitions

library

Specifies the document library.

documentNumber

Specifies the document number in the library.

filename

Specifies the extension of the file.

documentTypeName

Specifies the document type.

author

Specifies the author of the document.

creator

Specifies the creator of the document.

officialVersion

Specifies the version of the document that is marked as official.

currentVersion

Specifies the version of the document that is current.

versionDescription

Specifies the description of the document version.

fileSize

Specifies the size of the file.

acl

Specifies the access control list of the document.

Filter

Specifies how you want to filter items.

Syntax

```
<Filter>
<element type="FilterElement" minOccurs="0" maxOccurs="1"/>
```

FilterDate

Allows clients to filter on dates relative to Today, Tomorrow, ThisMonth, ThisWeek, ThisYear, and Yesterday.

Syntax

```
<FilterDate>
<restriction base="string">
<enumeration value="Today"/>
<enumeration value="Tomorrow"/>
<enumeration value="ThisMonth"/>
<enumeration value="ThisWeek"/>
<enumeration value="ThisYear"/>
<enumeration value="Yesterday"/>
```

FilterElement

The base object for a filter.

Syntax

```
<FilterElement>
<op type="FilterOp"/>
```

Definition

op

Specifies the operation applied to a field and value.

FilterEntry

Defines a filter. FilterEntry extends FilterElement (page 52).

Syntax

```
<FilterEntry>
<extension base="FilterElement">
<field type="string"/>
<custom type="Custom"/>
<value type="string"/>
<date type="FilterDate"/>
<mask type="string"/>
```

Definitions

field

Specifies to filter on a specific field. The field value is set in the value element.

custom

Specifies to filter on a specific custom field.

value

Specifies to filter on a specific value.

date

Specifies to filter on a relative date. For a relative date, the value element is plus or minus the relative date.

mask

Applies to the bitCare FilterOp. It is a bit operator.

FilterGroup

Groups a collection of FilterEntry (page 53) or FilterGroup (page 54) elements. FilterGroup uses And, Or, or Not to group elements.

Syntax

```
<FilterGroup>
<extension base="FilterElement"
<element type="FilterElement" minOccurs="1" maxOccurs="unbounded"/>
```

FilterOp

Lists valid operations on a filter.

Syntax

```
<FilterOp>
<enumeration value="and"/>
<enumeration value="or"/>
<enumeration value="not"/>
<enumeration value="eq"/>
<enumeration value="ne"/>
<enumeration value="gt"/>
<enumeration value="lt"/>
<enumeration value="gte"/>
<enumeration value="lte"/>
<enumeration value="contains"/>
<enumeration value="containsWord"/>
<enumeration value="begins"/>
<enumeration value="exists"/>
<enumeration value="notExist"/>
<enumeration value="isOf"/>
<enumeration value="isNotOf"/>
<enumeration value="fieldEqual"/>
<enumeration value="fieldGTE"/>
<enumeration value="fieldGT"/>
<enumeration value="fieldLTE"/>
<enumeration value="fieldLT"/>
<enumeration value="fieldNE"/>
<enumeration value="fieldDateEqual"/>
<enumeration value="bitCare"/>
<enumeration value="notContains"/>
```

Definitions

and

Specifies to gather a group of filters. In the example below, two filters are ANDed together. The following filter returns items if the subject contains "custom" and the creation date is greater than 10-06-2012:

```
<filter>
<element type="FilterGroup">
<op>and</op>
<element type="FilterEntry">
<op>contains</op>
<field>subject</field>
<value>custom</value>
</element>
<element type="FilterEntry">
<op>gt</op>
<field>created</field>
<value>2012-10-06T00:00:00Z</value>
</element>
</element>
</filter>
```

or

Specifies to gather a group of filters. In the example below, two filters are ORed together. The following filter returns items if the subject contains "custom" or the creation date is greater than 10-06-2012:

```
<filter>
<element type="FilterGroup">
<op>or</op>

<element type="FilterEntry">
<op>contains</op>
<field>subject</field>
<value>custom</value>
</element>
<element type="FilterEntry">
<op>gt</op>
<field>created</field>
<value>2012-10-06T00:00:00Z</value>
</element>
</element>
</filter>
```

not

Specifies how to gather a group of filters. In the example below, two filters are NOTed together. The following filter returns all items that do not match if the subject contains “custom” and the creation date is greater than 10-06-2012:

```
<filter>
<element type="FilterGroup">
<op>or</op>
<element type="FilterEntry">
<op>contains</op>
<field>subject</field>
<value>custom</value>
</element>
<element type="FilterEntry">
<op>gt</op>
<field>created</field>
<value>2012-10-06T00:00:00Z</value>
</element>
</element>
</filter>
```

eq

Specifies to return items with equal a value. The following example returns items with a created date equal to 2012-09-15T20:25:05Z:

```
<filter>
<element type="FilterEntry">
<op>eq</op>
<field>created</field>
<value>2012-09-15T20:25:05Z</value>
</element>
</filter>
```

ne

Specifies to return items not equal to the value. The following example returns items with a created date not equal to 2012-09-15T20:25:05Z:

```
<filter>
<element type="FilterEntry">
<op>eq</op>
<field>created</field>
<value>2012-09-15T20:25:05Z</value>
</element>
</filter>
```

gt

Specifies that the field has a value greater than the value specified. The following example returns items with a startDate greater than 2012-10-07T00:00:00Z:

```
<filter>
<element type="FilterEntry">
<op>gt</op>
<field>startDate</field>
<value>2012-10-07T00:00:00Z</value>
</element>
</filter>
```

lt

Specifies that the field has a value less than the value specified. The following example returns items with a startDate less than 2012-10-07T00:00:00Z:

```
<filter>
<element type="FilterEntry">
<op>lt</op>
<field>startDate</field>
<value>2012-10-07T00:00:00Z</value>
</element>
</filter>
```

gte

Specifies that the field has a value greater than or equal to the value specified. The following example returns items with a startDate greater than or equal to 2012-10-07T00:00:00Z:

```
<filter>
<element type="FilterEntry">
<op>gte</op>
<field>startDate</field>
<value>2012-10-07T00:00:00Z</value>
</element>
</filter>
```

lte

Specifies that the field has a value less than or equal to the value specified. The following example returns items with a startDate less than or equal to 2012-10-07T00:00:00Z:

```
<filter>
<element type="FilterEntry">
<op>lt</op>
<field>startDate</field>
<value>2012-10-07T00:00:00Z</value>
</element>
</filter>
```

contains

Specifies that the value is contained in the field. The following example returns items with a subject that contains “Today:”

```
<filter>
<element type="FilterEntry">
<op>contains</op>
<field>subject</field>
<value>Today</value>
</element>
</filter>
```

containsWord

Specifies that the value is contained in the field. The following example returns items with a subject that contains “Today:”

```
<filter>
<element type="FilterEntry">
<op>containsWord</op>
<field>subject</field>
<value>Today</value>
</element>
</filter>
```

begins

Specifies that the value begins with the value in the specified field. The following example returns items with a subject that begins with “Setting:”

```
<filter>
<element type="FilterEntry">
<op>begins</op>
<field>subject</field>
<value>setting</value>
</element>
</filter>
```

exists

Specifies items that have a specific element in the item.

notExist

Specifies items that do not have a specific element in the item.

isOf

If the value element can have more than one value (such as mail, task, and appointment), the isOf operation can be used for the filter operation. In the following example, items are returned if the item types are appointment or task:

```
<filter>
<element type="FilterEntry">
<op>isOf</op>
<field>@type</field>
<value>appointment task</value>
</element>
</filter>
```

isNotOf

If the value element can have more than one value (such as mail, task, and appointment), the isNotOf operation can be used for the filter operation. In the following example, items are returned if the item type is not an appointment:

```
<filter>
<element type="FilterEntry">
<op>isNotOf</op>
<field>@type</field>
<value>appointment</value>
</element>
</filter>
```

fieldEqual

Used with the relative date element in the filter. FieldEqual indicates that the date equals the relative date. For example, the following filter returns items with a startDate 1 day less than Today:

```
<filter>
<element type="FilterEntry">
<op>fieldEqual</op>
<field>startDate</field>
<value>-1</value>
<date>Today</date>
</element>
</filter>
```

fieldGTE

Used with the relative date element in the filter. fieldGTE indicates that the date is greater than or equal to the relative date. For example, the following filter returns items that are greater than or equal to the startDate of Today minus 1 day:

```
<filter>
<element type="FilterEntry">
<op>fieldGTE</op>
<field>startDate</field>
<value>-1</value>
<date>Today</date>
</element>
</filter>
```

fieldGT

Used with the relative date element in the filter. fieldGT indicates that the date is greater than the relative date. For example, the following filter returns items that are greater than the startDate of Today minus 1 day:

```
<filter>
<element type="FilterEntry">
<op>fieldGT</op>
<field>startDate</field>
<value>-1</value>
<date>Today</date>
</element>
</filter>
```

fieldLTE

Used with the relative date element in the filter. fieldLTE indicates that the date is less than or equal to the relative date. For example, the following filter returns items that are less or equal than the startDate of Today minus 3 days:

```
<filter>
<element type="FilterEntry">
<op>fieldLTE</op>
<field>startDate</field>
<value>-3</value>
<date>Today</date>
</element>
</filter>
```

fieldLT

Used with the relative date element in the filter. fieldLT indicates that the date is less than the relative date. For example, the following filter returns items that are less than the startDate of Today minus 5 days:

```
<filter>
<element type="FilterEntry">
<op>fieldLT</op>
<field>startDate</field>
<value>-5</value>
<date>Today</date>
</element>
</filter>
```

fieldNE

Used with the relative date element in the filter. fieldNE indicates that the date is not equal to the relative date:

```
<filter>
<element type="FilterEntry">
<op>fieldNE</op>
<field>startDate</field>
<value>-4</value>
<date>Today</date>
</element>
</filter>
```

fieldDateEqual

Used with the relative date element in the filter (similar to fieldEqual).

bitCare

Translates into an operator that means “Is Not Of” or “Does Not Include” or “Not Equal To”. In the following example, items that are not appointments are returned. The field specifies ItemType and the mask specifies the “Is Not Of” type Appointment.

```
<filter>
<element type="FilterEntry">
<op>bitCare</op>
<field>@type</field>
<value />
<mask>Appointment</mask>
</element>
</filter>
```

The following example has a value of Appointment. This translates into a filter where ItemType Equals Appointment.

```
<filter>
<element type="FilterEntry">
<op>bitCare</op>
<field>@type</field>
<value Appointment/>
<mask>Appointment</mask>
</element>
</filter>
```

notContains

Specifies that the value is not contained in the field. The following example returns items with a subject that does not contain “Today”.

```
<filter>
<element type="FilterEntry">
<op>notContains</op>
<field>subject</field>
<value>Today</value>
</element>
</filter>
```

From

Identifies the user that sent an item.

Syntax

```
<From>
<extension base="NameAndEmail"/>
<replyTo type="string"/>
```

Definitions

NameAndEmail

From extends NameAndEmail (page 80).

replyTo

Specifies the IMAP replyTo field.

GMTOffset

Contains the time zone offset (from UTC in seconds).

Syntax

```
<GMTOffset>
<restriction base="int">
<minExclusive value="-86400"/>
<maxExclusive value="86400"/>
```

Hour

Restricts the hour value between 0 and 23.

Syntax

```
<Hour>
<restriction base="unsignedByte">
<maxInclusive value="23"/>
```

Item

A base object that defines an item in GroupWise. Many objects extend Item.

Syntax

```
<Item>
<ref="id"/>
<ref="name"/>
<ref="version"/>
<ref="modified"/>
<name="changes" type="ItemChanges"/>
```

Definitions

id

Specifies a string that uniquely identifies the item. All characters, up to the first at symbol (@), uniquely identify the item. All characters after the @ symbol control access to the item.

name

Specifies the name of the item.

version

Specifies the version number of the item.

modified

Specifies when the item was last modified.

changes

Specifies the changes to the item.

ItemClass

Marks an item as Private on a send.

Syntax

```
<ItemClass>
<enumeration value="Public"/>
<enumeration value="Private"/>
```

ItemList

Contains a list of items.

Syntax

```
<ItemList>
<item type="Item" minOccurs="0" maxOccurs="unbounded"/>
<attribute name="offset"/>
<attribute name="count"/>
```

Definitions

item

Specifies a list of items.

offset

Specifies the offset for the anchor position of the cursor.

count

Specifies the number of items to retrieve.

ItemOptions

Contains miscellaneous GroupWise options for an item.

Syntax

```
<priority type="ItemOptionsPriority"/>
<expires type="dateTime"/>
<delayDeliveryUntil type="dateTime"/>
<concealSubject type="boolean"/>
<hidden type="boolean"/>
```

Definitions

priority

Specifies the priority of the item: high, standard, or low.

expires

Species the expiration date of the item. This means that the item is removed from the recipient's account on the expiration date.

delayDeliveryUntil

Specifies to delay the delivery of the item until the specified date.

concealSubject

Specifies to conceal the subject of sent items. The subject is visible only when the recipient opens the item.

hidden

Specifies to mark the item as hidden. The only way to retrieve hidden items is to add “hidden” to the view.

ItemOptionsPriority

Contains an enumeration that describes the priority of an item.

Syntax

```
<ItemOptionsPriority>
<enumeration value="High"/>
<enumeration value="Standard"/>
<enumeration value="Low"/>
```

ItemSecurity

Contains an enumeration for the security of an item. In GroupWise, the security field does not limit visibility.

Syntax

```
<ItemSecurity>
<enumeration value="Normal"/>
<enumeration value="Proprietary"/>
<enumeration value="Confidential"/>
<enumeration value="Secret"/>
<enumeration value="TopSecret"/>
<enumeration value="ForYourEyesOnly"/>
```

ItemSource

Contains an enumeration of the source of an item.

Syntax

```
<ItemSource>
<enumeration value="received"/>
<enumeration value="sent"/>
<enumeration value="draft"/>
<enumeration value="personal"/>
```

Definitions

received

Specifies that an item was received.

sent

Specifies an item that was sent from this mailbox to others.

draft

Specifies that an item is a work in progress. The item has not been sent yet.

personal

Specifies a personal or posted item that appears only in the sender′s mailbox.

ItemStatus

Contains an enumeration of events on an item.

Syntax

```
<name="ItemStatus">
<accepted type="boolean"/>
<completed type="boolean"/>
<delegated type="boolean"/>
<deleted type="boolean"/>
<private type="boolean"/>
<forwarded type="boolean"/>
<opened type="boolean"/>
<read type="boolean"/>
<replied type="boolean"/>
```

ItemThreading

Specifies discussion threading for a list of items. The view needs to contain the keyword "threading" and "default" to get back the threading elements.

Syntax

```
<ItemThreading>
<id type="string"/>
<parent type="string"/>
```

Definitions

id

Specifies the ID of the current item.

parent

Specifies the ID of the parent item.

Mail

Contains a base mail message object.

Syntax

```
<Mail>
<extension base="BoxEntry">
<element ref="subject"/>
<originalSubject type="string"/>
<subjectPrefix type="string"/>
<distribution type="Distribution"/>
<message type="MessageBody"/>
<attachments type="AttachmentInfo"/>
<options type="ItemOptions"/>
<link type="LinkInfo"/>
<hasAttachment type="boolean" default="0"/>
<size type="int" default="0"/>
<subType type="string"/>
<nntpOrImap type="boolean"/>
<smimeType type="SMimeOperation"/>
<checklist type="ChecklistInfo"/>
<xField type="string"/>
<originalId type="uid"/>
<archiveId type="string"/>
<threading type="ItemThreading"/>
<retentionModified type="dateTime"/>
<rssURL type="string"/>
<attribute name="stub" type="boolean"/>
```

Definitions

subject

Specifies the subject text. If the original is changed, this text becomes mySubect.

originalSubject

Specifies the original subject, if the subject is overridden.

subjectPrefix

Specifies a prefix to add to the beginning of the subject.

distribution

Specifies the From user and all recipients.

message

Specifies the body text of the message.

attachments

Specifies the list of attachments.

options

Specifies various GroupWise options.

link

Specifies the linkInfo for an item that is forwarding or replying to a message.

hasAttachment

Specifies if the message has attachments.

size

Specifies the size of the item. The size includes attachments.

subType

Specifies the subType on an item. Valid subTypes are a shared folder notification message (NGW.SHARED.FOLDER.NOTIFY) or a shared personal address book message (NGW.SHARED.PAB.NOTIFY).

nntpOrImap

Specifies if the item is an IMAP or NNTP item.

smime

Specifies if the item is signed or encrypted.

xField

The xField is a string value like "name=value" (for example, "xfieldName=xFieldValue"). The xField is stored in the item. The xFields are sent out via the GWIA to external users. There can be many xFields.

xFields are primarily used in the IMAP protocol and are not displayed by any of the GroupWise Clients.

originalId

Specifies the items original item ID if the item has been stubbed/archived.

archivedId

Specifies the archive ID if the item has been stubbed/archived.

threading

Specifies discussion threads.

retentionModified

Specifies the date and time that a significant or meaningful part of the item was modified.

RetentionModified is a little different than the retention time stamp. Suppose an item has a retention time stamp. If a user modifies the item by adding a personal attachment or some other significant change, retention software skips the item because it has already been retained. RetentionModified is now used to catch an item after the first retention if a significant change has occurred to the item.

rssURL

Specifies the RSS URL of the RSS item.

stub

Specifies if the item is a stubbed/archived item.

MessageBody

Contains the message body text. The message is encoded in Base64. Currently, there is only one MessageBody part. The MessageBody part is the text plain message body. The HTML message body is an attachment with the `text.htm` name. The HTML message body can have related part attachments. They are related if they come immediately after the `text.htm` attachment and they have a contentId element.

Syntax

```
<MessageBody>
<part type="MessagePart" maxOccurs="unbounded"/>
<attribute name="defaultPart" type="string"/>
```

MessagePart

Contains the parts of the message text. The data is always Base64.

Syntax

```
<MessagePart>
<extension base="base64Binary">
<attribute id type="uid"/>
<attribute contentId type="string"/>
<attribute contentType type="string"/>
<attribute length type="int"/>
<attribute offset type="int"/>
<attribute name="hash" type="string"/>
```

Definitions

id

(Optional) Specifies the ID of the message part.

contentId

Specifies the MIME content ID.

contentType

(Optional) Specifies the MIME content type.

length

(Optional) Specifies the size of the original data, not the Base64 size.

offset

(Optional) Specifies the offset from where to start reading. On large messageParts, the offset element can be used to read smaller chunks at one time.

hash

GroupWise clients creates the hash on new items. If a hash has already been created, it is returned in the attachments object. To compute the hash yourself, see http://www.burtleburtle.net/bob/hash/doobs.html. In Lookup3.c, hashlittle2() is called with a block size of 4096. The hash is 64bits. The hash string is a 16-byte hex string.

Minute

Restricts a minute to be between 0 and 59.

Syntax

```
<Minute>
<restriction base="unsignedByte">
<maxInclusive value="59"/
```

Month

Restricts months to be between 0 and 11.

Syntax

```
<Month>
<restriction base="unsignedByte"
<maxInclusive value="11"/>
```

NameAndEmail

Provides the unique identification of a GroupWise user.

Syntax

```
<NameAndEmail>
<ref="displayName"/>
<ref="e-mail"/>
<ref="uuid"/>
```

Definitions

displayName

Specifies the displayable name of the item.

e-mail

Specifies the email address of the item.

uuid

Specifies the unique identifier for a user. The ID never changes, even if the user moves or has multiple email addresses.

Note

Contains a GroupWise reminder note. Notes extends CalendarItem (page 38).

Syntax

```
<Note>
<extension base="CalendarItem">
<startDate type="date"/>
```

Definitions

startDate

Specifies the starting date for the note.

OccurrenceType

Contains an enumeration for the days of the week.

Syntax

```
<OccurrenceType>
<enumeration value="First"/>
<enumeration value="Second"/>
<enumeration value="Third"/>
<enumeration value="Fourth"/>
<enumeration value="Fifth"/>
<enumeration value="Last"/>
```

PhoneFlags

Contains phone message event flags.

Syntax

```
<PhoneFlags>
<called type="boolean"/>
<pleaseCall type="boolean"/>
<willCall type="boolean"/>
<returnedYourCall type="boolean"/>
<wantsToSeeYou type="boolean"/>
<cameToSeeYou type="boolean"/>
<urgent type="boolean"/>
```

PhoneMessage

Represents a GroupWise phone message.

Syntax

```
<PhoneMessage>
<extension base="Mail"
<caller type="string"/>
<company type="string"/>
<phone type="string"/>
<flags type="PhoneFlags"/>
```

Recipient

Identifies an address in an email distribution list.

Syntax

```
<Recipient>
<extension base="NameAndEmail"
<distType type="DistributionType"/>
<recipType type="RecipientType"/>
<recipientStatus type="RecipientStatus">
<ref="acceptLevel"/>
```

Definitions

distType

Specifies the TO, CC, and BC fields for the recipient.

recipType

Specifies the type of recipient: User, Group, etc.

recipientStatus

Specifies the tracking status of messages that were sent to other people. For example, the sender can track users who have opened and accepted appointments.

acceptLevel

Specifies the acceptLevel for free/busy time.

RecipientList

Contains a list of Recipient (page 85) elements.

Syntax

```
<RecipientList>
<recipient type="Recipient" minOccurs="0" maxOccurs="unbounded"/>
```

RecipientType

Contains an enumeration of the types of recipients.

Syntax

```
<RecipientType>
<enumeration value="User"/>
<enumeration value="Resource"/>
<enumeration value="PersonalGroup"/>
<enumeration value="SystemGroup"/>
<enumeration value="PersonalGroupMember"/>
<enumeration value="SystemGroupMember"/>
```

Status

Contains the corresponding response for each method response. The response status returns the success or failure of the request.

Syntax

```
<Status>
<ref="code"/>
<ref="description"/>
<info type="string"/>
<problems type="ProblemList"/>
```

Definitions

code

Specifies 0 if the method was successfully executed. If the value is greater than 0, there was a problem with the request.

description

Specifies an explanation of the error.

info

Not implemented at this time.

problems

Specifies any errors.

Task

Identifies a GroupWise task.

Syntax

```
<Task>
<extension base="CalendarItem">
<startDate type="date"/>
<dueDate type="date"/>
<assignedDate type="date"/>
<taskPriority type="string"/>
<completed type="boolean"/>
```

Definitions

startDate

Specifies the start date for the task.

dueDate

Specifies the due date for the task.

taskPriority

Specifies the GroupWise task priority. The task priority values can be from 1-999 or A-Z. It can also be a combination of alphanumeric and numeric. The alphanumeric values must appear before the numeric values. A1, A111, A999, Z1, Z111, and Z999 are acceptable values.

completed

Specifies if the task is completed.

Timezone

Describes a time zone.

Syntax

```
<Timezone>
<id" type="string"/>
<ref="description"/>
<daylight type="TimezoneComponent"/>
<standard type="TimezoneComponent"/>
```

Definitions

id

Specifies the internationally accepted time zone abbreviations.

description

Specifies a description of the time zone.

daylight

Specifies the month, hour, minutes, offset, and dayOfWeek for daylight saving time.

standard

Specifies the month, hour, minutes, offset, and dayOfWeek of the time zone.

TimezoneComponent

Defines a standard or daylight time start.

Syntax

```
<TimezoneComponent>
<name type="string"/>
<month type="Month"/>
<day type="Day"/>
<dayOfWeek type="DayOfWeek"/>
<hour type="Hour"/>
<minute type="Minute"/>
<offset type="GMTOffset"/>
```

Definitions

name

Specifies the name of the time zone.

month

Specifies the month the time zone starts.

day

Specifies the day the time zone starts.

dayOfWeek

Specifies the day of the week the time zone starts.

hour

Specifies the hour the time zone starts.

minute

Specifies the minute the time zone starts.

offset

Specifies the offset of the time zone from UTC.

TrustedApplication

Authenticates a user as a trusted application. GroupWise administrators create a trusted application name and key that allows applications to log in to the GroupWise system as any user. This is useful for applications that need to periodically log in to user mailboxes without a user's password.

Syntax

```
<TrustedApplication>
<extension base="Authentication">
<username type="string"/>
<name type="string"/>
<key type="string"/>
```

Definitions

username

Specifies the GroupWise user ID to log in under.

name

Specifies the trusted application name. The name is created by the GroupWise administrator.

key

Specifies the trusted application key. The key is generated using a program run by the GroupWise administrator.

uid

Defines the unique identifier. The unique identifier is an application-defined string that corresponds to an item. The string can optionally consist of two parts: a unique existence of the item and any instance information. The first and second parts of the ID are separated by the at symbol (@) and cannot contain the at symbol. To determine whether two items are the same item, your application should compare the first part of the ID (up to the at symbol) for both items.

Syntax

```
<uid>
<restriction base="string"
<pattern value="[^@]+(@*)?"/>
```

UUID

Contains the unique identifier on a given email system. It is used in various places to indicate that a sender or recipient is an internal user to the collaboration system, instead of an external user with an Internet address.

Syntax

```
<UUID>
<restriction base="string"/>
```

WeekDay

Contains an enumeration for the days of the week.

Syntax

```
<WeekDay>
<enumeration value="Sunday"/>
<enumeration value="Monday"/>
<enumeration value="Tuesday"/>
<enumeration value="Wednesday"/>
<enumeration value="Thursday"/>
<enumeration value="Friday"/>
<enumeration value="Saturday"/>
```

A Revision History

The following table lists changes made to the GroupWise Stubbing documentation (in reverse chronological order):

Release	Changes
November 2012	Reviewed and updated for use with GroupWise 2012.
February 2010	Added as an NDK component.