

# GroupWise® Software Developer Kit

## Web Services (SOAP)

October 2023

## **Legal Notices**

Copyright 1993 - 2023 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

---

# Contents

<b>About This Guide</b>	<b>11</b>
<b>1 Overview</b>	<b>13</b>
Information Resources	14
Overview	14
Web Services on a POA	15
Using ConsoleOne	15
Using the Command Line	15
Using a POA Startup File	15
SOAP Requests	16
Custom Fields on GroupWise Items	17
All Day Events and Time Zones	18
Debugging Tips	20
Trace Utilities	20
Free/Busy	20
Getting Items	24
Get Items Request	24
GroupWise Cursors	26
Stubbed Items	32
HTTP Connections	33
Message Bodies	34
Large Message Bodies	34
HTML Message Bodies	36
Proxy	38
Recurrence	38
Recurring Item Example	39
Recurring Daily Appointment Example	39
Recurring Item with a Rule Example	40
Redirection	40
Streaming Attachments	41
Trusted Application	45
Filters and Views	45
Filters	46
Filtering on Different Item Types	48
Views	57
ContactNotes on a Contact	66
Creation of a ContactNote	67
Modification of a ContactNote	68
Deletion of a ContactNote	69
Display Settings	69
Item Modification	70
Example	71

<b>2 Methods</b>	<b>73</b>
acceptRequest . . . . .	76
acceptShareRequest . . . . .	78
addItemRequest . . . . .	80
addItemsRequest . . . . .	81
addMembersRequest . . . . .	82
archiveRequest . . . . .	83
closeFreeBusySessionRequest . . . . .	86
completeRequest . . . . .	87
createCursorRequest . . . . .	88
createItemRequest . . . . .	90
createItemsRequest . . . . .	92
createJunkEntryRequest . . . . .	93
createNotifyRequest . . . . .	94
createProxyAccessRequest . . . . .	96
createSignatureRequest . . . . .	98
declineRequest . . . . .	99
delegateRequest . . . . .	100
destroyCursorRequest . . . . .	102
executeRuleRequest . . . . .	103
forwardRequest . . . . .	104
getAccountListRequest . . . . .	106
getAddressBookListRequest . . . . .	107
getArchiveItemsRequest . . . . .	109
getAttachmentRequest . . . . .	111
getCategoryListRequest . . . . .	113
getContactHistoryFilterRequest . . . . .	115
getCustomListRequest . . . . .	127
getDeltasRequest . . . . .	128
getDeltaInfoRequest . . . . .	131
getDiskSpaceUsageRequest . . . . .	133
getDocumentTypeListRequest . . . . .	134
getFolderRequest . . . . .	136
getFolderListRequest . . . . .	138
getFreeBusyRequest . . . . .	143
getItemRequest . . . . .	146
getItemsRequest . . . . .	148
getJunkEntriesRequest . . . . .	154
getJunkMailSettingsRequest . . . . .	156
getLibraryItemRequest . . . . .	158
getLibraryListRequest . . . . .	160
getMemberOfRequest . . . . .	161
getNotifyListRequest . . . . .	163
getProxyAccessListRequest . . . . .	165
getProxyListRequest . . . . .	167
getQuickMessagesRequest . . . . .	169
getRuleListRequest . . . . .	173
getSettingsRequest . . . . .	176
getSignaturesRequest . . . . .	184

getStringRequest . . . . .	186
getTimestampRequest . . . . .	187
getTimezoneListRequest . . . . .	189
getUnreadRequest . . . . .	191
getUserListRequest . . . . .	195
loginRequest . . . . .	197
logoutRequest . . . . .	202
markPrivateRequest . . . . .	203
markReadRequest . . . . .	204
markUnPrivateRequest . . . . .	205
markUnReadRequest . . . . .	206
modifyItemRequest . . . . .	207
modifyItemsRequest . . . . .	211
modifyJunkEntryRequest . . . . .	212
modifyJunkMailSettingsRequest . . . . .	213
modifyPasswordRequest . . . . .	214
modifyNotifyRequest . . . . .	215
modifyProxyAccessRequest . . . . .	216
modifySettingsRequest . . . . .	218
modifySignaturesRequest . . . . .	219
moveItemRequest . . . . .	220
moveItemsRequest . . . . .	222
positionCursorRequest . . . . .	224
purgeDeletedItemsRequest . . . . .	226
purgeRequest . . . . .	227
readCursorRequest . . . . .	228
removeCustomDefinitionRequest . . . . .	230
removeItemRequest . . . . .	232
removeItemsRequest . . . . .	233
removeJunkEntryRequest . . . . .	235
removeMembersRequest . . . . .	236
removeProxyAccessRequest . . . . .	237
removeNotifyRequest . . . . .	238
removeSignatureRequest . . . . .	239
replyRequest . . . . .	240
resendRequest . . . . .	242
resolveRequest . . . . .	248
restoreItemRequest . . . . .	249
retractRequest . . . . .	250
sendItemRequest . . . . .	252
setTimestampRequest . . . . .	254
startFreeBusySessionRequest . . . . .	256
streamedSearchRequest . . . . .	258
stblItemRequest . . . . .	262
unacceptRequest . . . . .	263
unarchiveRequest . . . . .	264
uncompleteRequest . . . . .	265
updateversionStatusRequest . . . . .	266

### 3 Schema Elements

269

Simple Objects . . . . .	270
acceptLevel . . . . .	271
code . . . . .	272
description . . . . .	273
displayName . . . . .	274
e-mail . . . . .	275
endDate . . . . .	276
gwTrace . . . . .	277
id . . . . .	278
modified . . . . .	279
name . . . . .	280
place . . . . .	281
recurrenceKey . . . . .	282
rights . . . . .	283
sequence . . . . .	284
session . . . . .	285
sid . . . . .	286
startDate . . . . .	287
subject . . . . .	288
uuid . . . . .	289
version . . . . .	290
Complex Objects . . . . .	291
AcceptLevel . . . . .	298
AccessControlListEntry . . . . .	299
AccessControlList . . . . .	300
AccessMiscRight . . . . .	301
AccessRight . . . . .	302
AccessRightChanges . . . . .	303
AccessRightEntry . . . . .	304
AccessRightList . . . . .	305
Accounts . . . . .	306
AccountFlags . . . . .	307
AddressBook . . . . .	309
AddressBookItem . . . . .	310
AddressBookList . . . . .	312
AdvancedInfo . . . . .	313
AgeAction . . . . .	315
Alarm . . . . .	316
Appointment . . . . .	317
AppointmentConflict . . . . .	318
AttachmentFlags . . . . .	319
AttachmentID . . . . .	320
AttachmentInfo . . . . .	321
AttachmentItemInfo . . . . .	322
Authentication . . . . .	324
BoxEntry . . . . .	325
CalendarFolderAttribute . . . . .	327
CalendarFolderFlags . . . . .	328
CalendarItem . . . . .	329
CalendarPublish . . . . .	330
CalendarPublishRange . . . . .	331
CalendarViewType . . . . .	332
CalHost . . . . .	333

Category	334
CategoryFlags	335
CategoryList	336
CategoryRefList	337
ChecklistInfo	338
ColumnSettings	339
ColumnType	340
CommentStatus	341
Contact	342
ContactFolder	344
ContactNote	345
ContactNotes	346
ContactRefList	347
ContactType	348
ContainerItem	349
ContainerRef	350
CursorSeek	351
Custom	352
CustomList	353
CustomType	354
Day	355
DayOfMonth	356
DayOfMonthList	357
DayOfWeek	358
DayOfYear	359
DayOfYearList	360
DayOfYearWeek	361
DaysOfYearWeekList	362
DelegatedStatus	363
DelegateeStatus	364
DeltaInfo	365
DiskSpaceUsage	366
DisplayFlags	367
DisplayFolderType	369
DisplaySettings	370
DisplaySettingsList	371
DisplaySettingsType	372
Distribution	373
DistributionType	374
Document	375
DocumentRef	377
DocumentType	379
DocumentTypeList	380
EmailAddressList	381
Execution	382
External	383
Filter	384
FilterDate	385
FilterElement	386
FilterEntry	387
FilterGroup	388
FilterOp	389
Folder	397
FolderACL	399
FolderACLEntry	400

FolderACLStatus . . . . .	401
FolderDisplaySettings . . . . .	402
FolderFlags . . . . .	403
FolderList . . . . .	404
FolderType . . . . .	405
FreeBusyBlock . . . . .	408
FreeBusyBlockList . . . . .	409
FreeBusyInfo . . . . .	410
FreeBusyInfoList . . . . .	411
FreeBusyStats . . . . .	412
FreeBusyUserList . . . . .	413
Frequency . . . . .	414
From . . . . .	415
FullName . . . . .	416
GeneralAccount . . . . .	417
GMTOffset . . . . .	418
Group . . . . .	419
GroupMember . . . . .	420
GroupMemberList . . . . .	421
Header . . . . .	422
Host . . . . .	423
Hour . . . . .	424
HTMLImages . . . . .	425
ImAddress . . . . .	426
ImAddressList . . . . .	427
IMAP . . . . .	428
IMAPFolder . . . . .	429
Item . . . . .	430
ItemChanges . . . . .	431
ItemClass . . . . .	432
ItemList . . . . .	433
ItemOptions . . . . .	434
ItemOptionsPriority . . . . .	435
ItemRef . . . . .	436
ItemRefList . . . . .	437
Items . . . . .	438
ItemSecurity . . . . .	439
ItemSource . . . . .	440
ItemSourceList . . . . .	441
ItemStatus . . . . .	442
ItemThreading . . . . .	443
JunkEntry . . . . .	444
JunkHandlingListType . . . . .	445
JunkMatchType . . . . .	446
Library . . . . .	447
LibraryList . . . . .	448
LinkInfo . . . . .	449
LinkType . . . . .	450
Mail . . . . .	451
MessageBody . . . . .	454
MessagePart . . . . .	455
MessageType . . . . .	456
MessageTypeList . . . . .	457
Minute . . . . .	458
ModifyItem . . . . .	459



Month	460
MonthList	461
MoveItem	462
NameAndEmail	463
Nickname	464
NNTP	465
NNTPFolder	466
NNTPSettings	467
Note	468
NotificationType	469
NotifyEntry	470
NotifyEntryChanges	471
NotifyList	472
OccurrenceType	473
OfficeInfo	474
Organization	475
Owner	476
PanelDisplaySettings	477
PanelList	479
PersonalInfo	480
PhoneFlags	482
PhoneList	483
PhoneMessage	484
PhoneNumber	485
PhoneNumberType	486
PictureData	487
PlainText	488
PollSettings	489
PostalAddress	490
PostalAddressList	491
PostalAddressType	492
ProblemEntry	493
ProblemList	494
Proxy	495
ProxyFolder	496
ProxyList	497
Query	498
QueryFolder	499
QueryTarget	500
QueryTargetList	501
RangeDays	502
Recipient	503
RecipientList	504
RecipientStatus	505
RecipientType	508
RecurrenceDateType	509
RecurrenceRule	510
ReferenceInfo	512
Resource	513
RetractType	514
ReturnNotification	515
ReturnNotificationOptions	516
Rights	517
Rule	518
RuleAction	519

RuleActionList	520
RuleActionType	521
RuleList	522
SendOptions	523
SendOptionsRequestReply	524
SendTotals	525
Server	526
ServerFlags	527
Settings	528
SettingsGroup	529
SettingsList	530
SharedBook	531
SharedFolder	532
SharedFolderNotification	533
SharedNotification	534
Signature	535
SignatureData	536
Signatures	537
SignatureSettings	538
SignatureType	539
SMimeOperation	540
Sort	541
SortType	542
SourceType	543
Status	544
StatusTracking	545
StatusTrackingFlags	546
StatusTrackingOptions	547
Subscribe	548
SystemFolder	549
Task	550
TextLines	551
Timezone	552
TimezoneComponent	553
TimezoneList	554
TransferFailedStatus	555
TrustedApplication	556
uid	557
UserContactFolder	558
UserInfo	559
UserList	561
UUID	562
VacationRule	563
Version	564
VersionEvent	566
VersionEventType	567
VersionStatus	568
View	569
ViewType	570
Visibility	571
WebAccessSettings	572
WeekDay	573

# About This Guide

GroupWise Web Services provides developers with server-side access to user mailboxes. This programmatic access allows you to read and write directly to users' mailboxes by using industry standards such as XML, SOAP, and HTTP.

---

**IMPORTANT:** Unless otherwise marked, the features in GroupWise Web Services work with GroupWise 8 and later versions.

---

This guide contains the following sections:

- ♦ [Chapter 1, "Overview," on page 13](#)
- ♦ [Chapter 2, "Methods," on page 73](#)
- ♦ [Chapter 3, "Schema Elements," on page 269](#)

## 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.



# 1 Overview

GroupWise Web Services provides developers with server-side access to user mailboxes. This programmatic access allows you to read and write directly to users' mailboxes by using industry standards such as XML, SOAP, and HTTP.

GroupWise Web Services communicates directly with a user's post office agent (POA), and GroupWise schemas define the methods and structures that are used in the conversation with the GroupWise POA.

The GroupWise Web Services schema definition file (WSDL) provides the tools you need to hook into integrated development environment (IDE) frameworks that support Web Services.

For the tools you need to hook into IDE frameworks that support Web Services, GroupWise Web Services also uses the GroupWise Web Services Definition Language (WSDL).

In addition to GroupWise Web Services, GroupWise gives you access to events or actions that occur on a GroupWise user's mailbox. This extension to GroupWise Web Services is called GroupWise Events. GroupWise Events is a Web service that allows you to programmatically configure and retrieve specific GroupWise events that have occurred on a user's mailbox.

GroupWise Events and GroupWise Web Services should be used together. To learn more about GroupWise Events, see [GroupWise SDK: Web Services Events](#).

GroupWise Web Services and GroupWise Web Services Events are available on the following platforms:

- ♦ Linux
- ♦ NetWare (Not available for GroupWise 2012 and later versions)
- ♦ Windows

This section contains the following topics:

- ♦ ["Information Resources" on page 14](#)
- ♦ ["Overview" on page 14](#)
- ♦ ["Web Services on a POA" on page 15](#)
- ♦ ["SOAP Requests" on page 16](#)
- ♦ ["Custom Fields on GroupWise Items" on page 17](#)
- ♦ ["All Day Events and Time Zones" on page 18](#)
- ♦ ["Debugging Tips" on page 20](#)
- ♦ ["Free/Busy" on page 20](#)
- ♦ ["Getting Items" on page 24](#)
- ♦ ["HTTP Connections" on page 33](#)
- ♦ ["Message Bodies" on page 34](#)
- ♦ ["Proxy" on page 38](#)

- ♦ “Recurrence” on page 38
- ♦ “Redirection” on page 40
- ♦ “Streaming Attachments” on page 41
- ♦ “Trusted Application” on page 45
- ♦ “Filters and Views” on page 45
- ♦ “ContactNotes on a Contact” on page 66
- ♦ “Display Settings” on page 69
- ♦ “Item Modification” on page 70

## Information Resources

Before using GroupWise Web Services, you should be familiar with XML, SOAP, and HTTP concepts. You can find more information with the following links:

- ♦ [GroupWise Documentation](#)
- ♦ [HTTP page \(https://www.w3.org/Protocols/Specs.html\)](https://www.w3.org/Protocols/Specs.html)
- ♦ SOAP
  - ♦ [Part 0 \(https://www.w3.org/TR/soap12-part0/\)](https://www.w3.org/TR/soap12-part0/)
  - ♦ [Part 1 \(https://www.w3.org/TR/soap12-part1/\)](https://www.w3.org/TR/soap12-part1/)
  - ♦ [Part 2 \(https://www.w3.org/TR/soap12-part2/\)](https://www.w3.org/TR/soap12-part2/)
- ♦ XML Schema Specifications
  - ♦ [Overview \(https://www.w3.org/XML/Schema\)](https://www.w3.org/XML/Schema)
  - ♦ [Part 0 \(https://www.w3.org/TR/xmlschema-0/\)](https://www.w3.org/TR/xmlschema-0/)
  - ♦ [Part 1 \(https://www.w3.org/TR/xmlschema-1/\)](https://www.w3.org/TR/xmlschema-1/)
  - ♦ [Part 2 \(http://www.w3.org/TR/xmlschema-2/\)](http://www.w3.org/TR/xmlschema-2/)
- ♦ [XML \(https://www.w3.org/XML/\)](https://www.w3.org/XML/)
- ♦ [WSDL \(https://www.w3.org/TR/wsdl\)](https://www.w3.org/TR/wsdl)

## Overview

The GroupWise Web Services object model and methods are defined in the XML schemas. Currently, the GroupWise schemas include:

- ♦ `types.xsd`, which defines GroupWise Web Services.
- ♦ `methods.xsd`, which defines GroupWise Web Services.
- ♦ `events.xsd`, which defines GroupWise Events and which references `types.xsd`.

The GroupWise WSDL and schemas are available in the GroupWise Web Services SDK.

We provide you with the schemas and WSDL files. You then decide the specific development framework you want to use to build GroupWise applications. Each development framework compiles its own library with the GroupWise schemas and WSDL. Different frameworks have different levels of support for Web Services. This means that each generated library could have a different set of bugs.

We tested the SOAP libraries with Java up to and including version 7, and with Microsoft .NET up to and including version 4.5.

## Web Services on a POA

GroupWise Web Services uses SOAP to communicate with the GroupWise POA.

For more details on enabling SOAP on a POA, see [“Supporting SOAP Clients”](#) in the *GroupWise Administration Guide*.

The protocol is enabled at the post office. You can enable SOAP in the following three ways:

- ♦ [“Using ConsoleOne” on page 15](#)
- ♦ [“Using the Command Line” on page 15](#)
- ♦ [“Using a POA Startup File” on page 15](#)

### Using ConsoleOne

To enable SOAP by using ConsoleOne:

- 1 In ConsoleOne, browse to and right-click the POA object, then click **Properties**.
- 2 Click **GroupWise > Agent Settings**.
- 3 Select **Enable SOAP**.

### Using the Command Line

To enable SOAP through the command line, add `/soap-enabled` to the command line, so that the line reads:

```
gwpoa @pol.POA /soap-enabled
```

### Using a POA Startup File

To enable SOAP using the startup file:

- 1 Open the POA startup file and find the following line:

```
; /soap-[disabled|Enabled]
```

- 2 In the configuration file, change the line to read:

```
/soap-Enabled
```

# SOAP Requests

GroupWise POAs accept SOAP requests using a URL. Request URLs look like the following:

`http://[IP address]:port/soap` or `https://[IP address]:port/soap`

## **http**

HTTP is the GroupWise Web Service transport protocol and appears as it does when accessing a URL in a browser.

## **https**

HTTPS is the secure GroupWise Web Service transport protocol. In order to use HTTPS (SSL), each POA needs to have SSL connections enabled for SOAP.

## **IP address**

The IP address is the DNS name or the actual IP address of the POA.

## **Port**

The default port for GroupWise Web Services (SOAP) is 7191. This port can be changed.

## **/soap**

Tells the POA to process the HTTP request as a SOAP request. For example, you could access a test POA with SOAP by using `http://172.16.5.18:7191/soap`.

In the HTTP headers, you should be familiar with the `soapaction` and `contenttype` headers.

## **soapaction**

The `soapaction` HTTP header field can be used to indicate the intent of the SOAP HTTP request. Soapaction is required. However, it does not need to have a value.

## **contenttype**

The `contenttype` HTTP header field indicates the media type of the entity body that was sent to the recipient. The value of the `contenttype` header should be `text/xml` or `application/soap+xml`. This field value is required.

Requests support only a charset of UTF-8. Responses return only a charset of UTF-8. Any other charset that is passed to the POA is ignored.

The following is an example of an HTTP header:

```
<HTTPHeaders>
<soapaction>"fRequest"</soapaction>
<content-type>text/xml; charset="UTF-8"</content-type>
<user-agent>SOAP Toolkit 3.0</user-agent>
<host>172.16.5.18:8080</host>
<content-length>560</content-length>
<connection>Keep-Alive</connection>
<cache-control>no-cache</cache-control>
<pragma>no-cache</pragma>
</HTTPHeaders>
```

The following is an example of a HTTP response header:



```

<HTTPHeaders>
<date>Thu, 31 Aug 2012 17:51:26 GMT</date>
<server>Windows GroupWise POA 12.0.0</server>
<content-type>text/xml; charset=utf-8</content-type>
<content-length>71542</content-length>
<pragma>no-cache</pragma>
</HTTPHeaders>

```

## Custom Fields on GroupWise Items

If you want to add application-specific data to items in the GroupWise store, you can do this with custom fields. For example, if application X wanted to track the items it created in a user's data store, application X could add a custom field with a value of X to all items it creates. Later, the application can use filters to find application X's items.

When a custom field is added to an item, the new custom field definition is added to the GroupWise post office database. If the post office database has too many custom fields, POA performance can be reduced. We recommend that the number of custom fields be kept to a minimum and that applications use no more than two or three custom fields.

To list the custom field definitions on a post office, call `getCustomListRequest`, which returns the list of custom fields without any associated values.

To remove a custom field definition in the post office database, call `removeCustomDefinitionRequest`. Because the custom field needs to be removed from the items before the definition can be removed from the post office guardian database, this method can take some time to process.

You can add a custom fields to new or existing items, You can also modify existing custom fields. Custom fields can be string, date, numeric, or binary data.

To create custom fields on new items, you need to specify the custom fields in the `sendItemRequest` (or `createItemRequest`) as in the following example:

```

<ns0:sendItemRequest>
<ns0:item xmlns:ans1="http://schemas.novell.com/2005/01/GroupWise/
types" xsi:type="ans1:Mail">
<ans1:version>0</ans1:version>
<ans1:customs>
  <ans1:custom type="String">
    <ans1:field>customField1</ans1:field>
    <ans1:value>StringValue</ans1:value>
  </ans1:custom>
</ans1:customs>
...

```

You can add a custom field to an existing posted or distributed item with the `ModifyItemRequest`.

```

<ns0:modifyItemRequest>
<ns0:id>48F60187.domain1.pol.100.16A6163.1.138B.1@1:7.domain1.pol.100.0.1.
0.1@16</ns0:id>
<ns0:updates>
<ans1:update xmlns:ans1="http://schemas.novell.com/2005/01/GroupWise/
types" xsi:type="ans1:Mail">
<ans1:version>0</ans1:version>
<ans1:customs>
<ans1:custom type="String">
<ans1:field>customField2</ans1:field>
<ans1:value>StringValue2</ans1:value>
</ans1:custom>
</ans1:customs>
</ans1:update>
</ns0:updates>
...
</ns0:modifyItemRequest>

```

You can also modify an existing custom field by passing in the name of the custom field with the new value.

```

<ns0:modifyItemRequest>
<ns0:id>48F60187.domain1.pol.100.16A6163.1.138B.1@1:7.domain1.pol.100.0.1.
0.1@16</ns0:id>
<ns0:updates>
<ans1:update xmlns:ans1="http://schemas.novell.com/2005/01/GroupWise/
types" xsi:type="ans1:Mail">
<ans1:version>0</ans1:version>
<ans1:customs>
<ans1:custom type="String">
<ans1:field>customField1</ans1:field>
<ans1:value>New String Value</ans1:value>
</ans1:custom>
</ans1:customs>
</ans1:update>
</ns0:updates>
...
</ns0:modifyItemRequest>

```

## All Day Events and Time Zones

GroupWise all day events can cause problems if applications need to apply the sender's time zone to the start and end dates.

### Problem

An application displays an all day event on the wrong day.

### Solution

GroupWise clients set the startDate and endDate to midnight of the user's time zone. As the item is saved to the user's database, all times are converted to UTC. This causes problems in determining the actual date of the all day event because the date and time on the event shifts. The shift depends the time zone of the GroupWise clients.

As an example, suppose a GroupWise client sends an all day event on 9/26/2012 with a time zone of Mountain Daylight Time. The output for this all day event is as follows:

```
<getItemResponse>
<item type="Appointment">
...
<subject>All Day Event</subject>
<startDate>2012-09-26T06:00:00Z</startDate>
<endDate>2012-09-27T06:00:00Z</endDate>
<startDay>2012-09-26</startDay>
<endDay>2012-09-27</endDay>
<allDayEvent>1</allDayEvent>
<timezone>
<daylight>
<name>Mountain Daylight Time</name>
<month>4</month>
<dayOfWeek occurrence="First">Sunday</dayOfWeek>
<hour>2</hour>
<minute>0</minute>
<offset>-21600</offset>
</daylight>
<standard>
<name>Mountain Standard Time</name>
<month>10</month>
<dayOfWeek occurrence="Last">Sunday</dayOfWeek>
<hour>2</hour>
<minute>0</minute>
<offset>-25200</offset>
</standard>
</timezone>
</item>
```

In the output above, the startDate has a value of <startDate>2012-09-26T06:00:00Z</startDate>. The time is 6:00 a.m. because GroupWise saves the time in UTC. UTC is 6 hours ahead of MDT in our example.

In order to determine the real date of the all day event, you need to change the startDate and endDate to midnight by using a time zone to represent the sender's time zone. An application adds the sender's time zone offset to the startDate and endDate values.

Using the example above, this appears as follows:

```
<startDate>2012-09-26T06:00:00Z</startDate> + <offset>-21600</offset> =
<startDate>2012-09-26T00:00:00Z</startDate>
```

The adjusted startDate is midnight. The endDate needs the same adjustment.

AllDayEvents returns a startDate, startDay, endDate, and endDay. The startDate and endDate have times associated with the dates, as follows:

```
<startDate>2012-09-26T06:00:00Z</startDate>
```

StartDay and endDay do not have time associated with the dates:

```
<startDay>2012-09-26</startDay>
```

Use the elements that don't return a time. If an all day event does not interact with other GroupWise or third-party clients, the startDay and endDay elements can be used. Otherwise, you need to use the adjusted startDate and endDate elements.

## Debugging Tips

Capturing and viewing SOAP/XML data on the wire is a good way to debug problems. There are two ways to capture SOAP/XML data:

- ♦ Microsoft provides a debugging trace tool with the SOAP Toolkit 3. It allows you to view the SOAP and XML requests and responses between clients and servers. However, the XML data can be viewed only if SSL is disabled. To download this trace utility, see [Microsoft's Web site \(http://www.microsoft.com/downloads/details.aspx?FamilyId=C943C0DD-CEEC-4088-9753-86F052EC8450&displaylang=en#overview\)](http://www.microsoft.com/downloads/details.aspx?FamilyId=C943C0DD-CEEC-4088-9753-86F052EC8450&displaylang=en#overview). This trace utility executable is called MsSoapT3.exe. The toolkit is a precursor to .NET. The trace utility runs only on Windows, but there are other trace utilities that accomplish the same thing.
- ♦ You can also set the gwTrace element in each request to True. If gwTrace is True, the request and response are written to the POA log directory. This is a good way to view request and responses that are encrypted with SSL. The files take on the format of datexml.001. For example, a request response might have a file name of 1006xml.001. It is a good idea to clean up the temporary files.

To view the GroupWise Web Service method calls in the POA log files, change the POA log level to Diagnostic. The user and the method are logged this way. If the POA is in normal or verbose mode, only the login and logout methods are written to the log files.

## Trace Utilities

The following utilities are useful for monitoring the data flowing on a TCP connection:

- ♦ tcpmon (<http://code.google.com/p/tcpmon>)
- ♦ TcpTrace (<http://www.pocketsoap.com/tcpTrace>)

## Free/Busy

Free/busy information is a real-time view of calendar free/busy time for a set of users. The free/busy time is returned in blocks of time. Each block of time has an associated acceptLevel of free, busy, tentative, or out of the office.

There are three methods used in retrieving free/busy time:

- ♦ [startFreeBusySessionRequest](#), which creates a server-side object to gather free/busy information for a set of users with a start and end date.
- ♦ [closeFreeBusySessionRequest](#), which destroys the server-side object and should be called when a free/busy search is complete.
- ♦ [getFreeBusyRequest](#), which retrieves the free/busy blocks for the users provided in the startFreebusySessionRequest.

Because of the real-time lookup, it might take time to gather free/busy blocks for users on different post offices. In the `getFreeBusyResponse`, there is a `freebusystats` element, which returns the total number of users in the free/busy call. It also returns the number of users that have responded with free/busy information and the number of outstanding users that have not returned busy/free information yet. You might need to call `getFreeBusyRequest` more than one time to retrieve all the free/busy time for the list of users. If a user has proxy rights to a user, the subject is returned in the free/busy block.

In the following example, `startFreeBusySessionRequest` provides a list of users and a start and end date for the free/busy search. The `startFreeBusySessionResponse` returns a `freeBusySessionId`, which identifies the server-side free/busy. The `freeBusySessionId` needs to be passed in as a parameters to both the `getFreeBusyRequest` and the `closeFreeBusySessionRequest`.

```
<startFreeBusySessionRequest>
  <users>
    <user>
      <displayName>u2</displayName>
      <email>u2@phantom.com</email>
      <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
    </user>
    <user>
      <displayName>FIRST01010000 LAST01010000</displayName>
      <email>USER01010000@phantom.com</email>
      <uuid>07B4A660-18CA-0000-B989-BA00EB000D00</uuid>
    </user>
  </users>
  <startDate>20121009T060000Z</startDate>
  <endDate>20121030T060000Z</endDate>
</startFreeBusySessionRequest>

<startFreeBusySessionResponse>
  <freeBusySessionId>4250</freeBusySessionId>
  <status>
    <code>0</code>
  </status>
</startFreeBusySessionResponse>
```

In the following request, one user out of two has responded. The `freeBusyInfo` element returns the blocks of free/busy time for the one user. You might want to wait up to a minute to call `getFreeBusyRequest` to get the free/busy time for the second user.

```
<getFreeBusyRequest>
  <freeBusySessionId>4250</freeBusySessionId>
</getFreeBusyRequest>
```

```

<getFreeBusyResponse>
  <freeBusyStats>
    <responded>1</responded>
    <outstanding>1</outstanding>
    <total>2</total>
  </freeBusyStats>
  <freeBusyInfo>
    <user>
      <email>u2@phantom.com</email>
      <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
      <blocks>
        <block>
          <startDate>2012-10-09T06:00:00Z</startDate>
          <endDate>2012-10-09T14:00:00Z</endDate>
          <acceptLevel>Busy</acceptLevel>
        </block>
        ...
        <block>
          <startDate>2012-10-27T23:00:00Z</startDate>
          <endDate>2012-10-30T07:00:00Z</endDate>
          <acceptLevel>OutOfOffice</acceptLevel>
        </block>
      </blocks>
    </user>
  </freeBusyInfo>
  <status>
    <code>0</code>
  </status>
</getFreeBusyResponse>

```

To destroy the server-side free/busy session, call `closeFreeBusySessionRequest`.

```

<closeFreeBusySessionRequest>
  <freeBusySessionId>4250</freeBusySessionId>
</closeFreeBusySessionRequest>

<closeFreeBusySessionResponse>
  <status>
    <code>0</code>
  </status>
</closeFreeBusySessionResponse>

```

Starting with GroupWise 8.0 SP3, external users are not included in the list of free/busy users. The `startFreeBusySessionRequest` returns the users that are external in the `problems` element.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SOAP-ENV:Envelope
  xmlns:SOAPSDK1="http://www.w3.org/2001/XMLSchema"
  xmlns:SOAPSDK2="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAPSDK3="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header SOAP-ENV:encodingStyle="">
    <session>XVNj1FPCVRwvvrED</session>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body SOAP-ENV:encodingStyle="">
    <startFreeBusySessionRequest>
      <users>
        <user>
          <displayName>Preston Stephenson</displayName>
          <email>Preston.Stephenson@myhost.com</email>
        </user>
        <user>
          <displayName>Preston Stephenson</displayName>
          <email>pstephenson@prestons.provo.novell.com</email>
          <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
        </user>
      </users>
      <startDate>20110119T070000Z</startDate>
      <endDate>20110209T070000Z</endDate>
    </startFreeBusySessionRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAPSDK3="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAPSDK2="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAPSDK1="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Header SOAP-ENV:encodingStyle=""/>
  <SOAP-ENV:Body SOAP-ENV:encodingStyle="">
    <startFreeBusySessionResponse>
      <freeBusySessionId>154702</freeBusySessionId>
      <status>
        <code>0</code>
        <problems>
          <entry>
            <element>Preston.Stephenson@myhost.com</element>
            <code>53505</code>
            <description>User not on post office</description>
          </entry>
        </problems>
      </status>
    </startFreeBusySessionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

---

**NOTE:** The ProblemEntry only has one element entry. The value that is put in the element entry is from the user element from the request. The order of the value to add is the email, displayName, then uuid element values.

---

Starting in GroupWise 2012, the external email addresses are dropped by default. There are two optional attributes that can be put on the FreeBusyUserList element: external and externalGroupWise.

If 1 or true is passed for the value of the attribute, the external email address is passed in the free/busy search.

The external attribute passes in all external users. The externalGroupWise attribute only passes GroupWise users that are in external domains.

## Getting Items

There are different ways to retrieve items from a GroupWise container. Each of these methods has an input parameter of a container. A container is an abstract wrapper of GroupWise items. A container can be a GroupWise folder, address book, or library.

There are special strings to use as the container ID. The folders attribute can be used to search over all folders in the user's mailbox. (This searches all of the folders with the exception of shared folders shared with the user, query folders and the Trash folder.) The books attribute can be used to search over all personal address books.

This section contains the following sections:

- ♦ [“Get Items Request” on page 24](#)
- ♦ [“GroupWise Cursors” on page 26](#)
- ♦ [“Stubbed Items” on page 32](#)

## Get Items Request

[getItemsRequest](#) can retrieve items from folders, query folders, shared folders, personal address books, the GroupWise Address Book, libraries, document versions, and document version events.

---

**NOTE:** If you are getting more than 20 items, you should use the cursor request, as described in [GroupWise Cursors](#).

---

For GroupWise 2012 and later:

There are times when you have an item id, but you don't know what container the item is in. You can use [getItemsRequest](#) to return the item. You pass the id value in an <item> element of the <items> element. You pass `folders` or `books` as the container.



```

<getItemsRequest>
  <container>folders</container>
  <view>id status subject startDate created</view>
  <items>
    <item>4E92DD25.domain.P01.100.1776172.1.454BF.1</item>
  </items>
</getItemsRequest>

```

If you have a sid, you can pass that. (You must also pass sid in the view, so the code knows the value is sid and not an id.)

```

<getItemsRequest>
  <container>folders</container>
  <view>id sid status subject startDate created</view>
  <items>
    <item>283839</ans1:item>
  </items>
</getItemsRequest>

```

On folder items, using `folders` as the container, the code cannot find the item in a shared folder shared with the user. On personal address books, the code can find the item even in personal address books shared with the user.

When you have events, the event sometimes does not include what container the item is in. There is a special formatted id you can use to quickly read the item. For example, if you have the following event:

```

event>ItemModify</event>
  <id>4E9D1539.domain.P01.100.1776172.1.46167.1</id>
  <sid>287079</sid>
  <timeStamp>2011-10-18T12:16:24Z</timeStamp>
  <field>MessageBody Subject</field>
  <key>test</key>
  <uid>868965</uid>
  <type>Mail</type>
</event>

```

Then you combine the `<id>` and the `<uid>` values to generate an id:

```
4E9D1539.domain.P01.100.1776172.1.46167.1@Event:868965
```

You pass this id in the `<item>` element. The code can quickly verify the user has the right to read the item.

---

**NOTE:** The event record must still exist in this case. The code uses the event record to verify that the user has access to the item. If the event record is already deleted, this method of getting the item fails.

---

```

<getItemsRequest>
  <container>folders</container>
  <view>id status subject startDate created</view>
  <items>
    <item>4E9D1539.domain.P01.100.1776172.1.46167.1@Event:868965</item>
  </items>
</getItemsRequest>

```

## GroupWise Cursors

GroupWise cursors provide you with a method to retrieve large amounts of container data in chunks. Cursors can be used to retrieve a subset of items with subsequent methods instead of getting all the items in one response. For example, suppose a user has 1,000 items in a folder and you retrieve 100 items per call. To retrieve all 1,000 items would mean calling a method 10 different times.

Cursors also provide you with the ability to read items from the start or end of a list. For example, there is a container with 1000 items. Item 1 is the start of the list and item 1000 is the end of the list. You can use cursors to read items from the start to the end of the list (item 1 to item 1000), or you can use cursors to read items from the end to the start of the list (item 1000 to item 1).

GroupWise cursors require the use of several methods. The methods are:

- ♦ [createCursorRequest](#), which creates a cursor on the server where the new cursor is a pointer to the container data.
- ♦ [readCursorRequest](#), which reads the data from the container.
- ♦ [positionCursorRequest](#), which sets the position of the cursor on the server.
- ♦ [destroyCursorRequest](#), which removes the cursor from the server.

A cursor has an anchor position in the list of items. The anchor position can be the start of the list, the current position in the list, or the end of the list. The [readCursorRequest](#) and [positionCursorRequest](#) method can position the anchor in the list. For example, there is another container with 1,000 items, with item 1 at the start position and item 1,000 at the end position. As you read items by calling [readCursorRequest](#), the cursor position moves. If you read 100 items with [readCursorRequest](#), the anchor position of the cursor is 101. The next time you call [readCursorRequest](#), reading starts at item 101 if the position element is `current`.

[readCursorRequest](#) has a forward element. If the forward element is `True`, [readCursorRequest](#) reads from the starting anchor position to the ending anchor position. If the forward element is `False`, [readCursorRequest](#) reads from the ending anchor position to the starting anchor position.

Cursor data can become stale. For example, you create a cursor that points to a folder. If an item is added to the folder after the cursor is created, it does not return the new item by calling [readCursorRequest](#). You need to create a cursor, read the data, and then close or destroy the cursor.

Consider the following when working with GroupWise cursors:

- ♦ You can create more than one cursor per logged-in user.

---

**NOTE:** Do not create a large number of cursors per user, because it creates overhead that might degrade your system's performance.

---

- ♦ You can use cursors on folders, shared folders, personal address books, and the GroupWise Address Book.
- ♦ You cannot use cursors on contact folders or query folders.
- ♦ You cannot use cursors to get documents from a library.

For example, you cannot use cursors to get documents from the Authored Library or the Default Library.

Using cursors to read the Documents folder returns document references only; it does not return the documents themselves. Use `streamedSearchRequest` on document libraries in the place of cursors.

If a container does not support cursors, the following error is returned:

```
<createCursorResponse>
<status>
<code>59916</code>
<description>The method called is not supported.</description>
</status>
</createCursorResponse>
```

## Cursor Examples

This section contains the following examples:

- ♦ [“readCursorRequest \(forward\)” on page 27](#)
- ♦ [“readCursorRequest \(backward\)” on page 29](#)
- ♦ [“positionCursorRequest \(backward\)” on page 30](#)
- ♦ [“createCursorRequest” on page 31](#)

### readCursorRequest (forward)

In the following example, `readCursorRequest` returns items in order of oldest to newest. The container is the Mailbox. The forward element is `True`, and the position element is `current`. Five items are returned because the count element is set to 5.

The first call to `readCursorRequest` returns items 1 to 5.

```
<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1186833449</cursor>
  <forward>true</forward>
  <position>current</position>
  <count>5</count>
</readCursorRequest>

<readCursorResponse>
  <items>
    <item>
      <id>450AB7C1.domain1.pol.100.16E3837.1.FD0.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-15T22:23:28Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>1</subject>
    </item>
```

```

    . . .items 2, 3, and 4
    <item>
      <id>450AB7EA.domain1.pol.100.16E3837.1.FD8.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-15T22:23:28Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>5</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</readCursorResponse>

```

The next call to readCursorRequest returns items 6-10.

```

<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1186833449</cursor>
  <forward>true</forward>
  <position>current</position>
  <count>5</count>
</readCursorRequest>

<readCursorResponse >
  <items>
    <item>
      <id>450AD389.domain1.pol.100.16E3837.1.1002.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:27Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>6</subject>
    </item>
    . . .items 7, 8, and 9
    <item>
      <id>450AD3A8.domain1.pol.100.16E3837.1.100A.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:29Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>10</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</readCursorResponse>

```

## readCursorRequest (backward)

In the following example, readCursorRequest returns items in order of newest to oldest. The container is the Mailbox. The forward element is False, and the position element is end. Five items are returned because the count element is set to 5.

The first call to readCursorRequest returns items 25 to 21.

```
<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1186833450</cursor>
  <forward>false</forward>
  <position>end</position>
  <count>5</count>
</readCursorRequest>

<readCursorResponse>
  <items>
    <item>
      <id>450AD40A.domain1.pol.100.16E3837.1.1028.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:36Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>25</subject>
    </item>
    . . .items 24, 23, and 22
    <item>
      <id>450AD3F2.domain1.pol.100.16E3837.1.1020.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:34Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>21</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</readCursorResponse>
```

The next call to readCursorRequest returns items 20 to 16.

```
<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1186833450</cursor>
  <forward>false</forward>
  <position>current</position>
  <count>5</count>
</readCursorRequest>
```

```

<readCursorResponse>
  <items>
    <item>
      <id>450AD3EB.domain1.pol.100.16E3837.1.101E.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:34Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>20</subject>
    </item>
    . . .items 19, 18, and 17
    <item>
      <id>450AD3D4.domain1.pol.100.16E3837.1.1016.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:32Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>16</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</readCursorResponse>

```

## positionCursorRequest (backward)

---

**NOTE:** The positionCursorRequest method can produce unexpected results. Instead of using this method, we recommend that you use readCursorRequest to read from the beginning or the end.

---

In the following example, [positionCursorRequest](#) positions the anchor at the end of the list with an offset of 5. The list end is item 25. Taking into account the offset from the end of the list, the anchor position is at item 20. readCursorRequest moves backward in the list because the forward element is False, and the position element is current, which means to start the read at the current anchor position of 20.

```

<positionCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1186833452</cursor>
  <seek>end</seek>
  <offset>5</offset>
</positionCursorRequest>

<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1186833452</cursor>
  <forward>false</forward>
  <position>current</position>
  <count>5</count>
</readCursorRequest>

```

```

<readCursorResponse>
  <items>
    <item>
      <id>450AD3EB.domain1.pol.100.16E3837.1.101E.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:34Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>20</subject>
    </item>
    . . .items 19, 18, and 17
    <item>
      <id>450AD3D4.domain1.pol.100.16E3837.1.1016.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:32Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>16</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</readCursorResponse>

```

## createCursorRequest

The following example demonstrates creating a cursor with a filter and a view by calling [createCursorRequest](#). The view specifies the elements that should be returned for each item in the request. The filter narrows the items that are read in the user's database. The following filter indicates to read items with a created date that is greater than 2012-08-13.

```

<createCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>modified subject</view>
  <filter>
    <element type="FilterEntry">
      <op>gt</op>
      <field>created</field>
      <value>2012-08-13T00:00:00Z</value>
    </element>
  </filter>
</createCursorRequest>

```

```

<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1188933747</cursor>
  <forward>true</forward>
  <position>start</position>
  <count>5</count>
</readCursorRequest>

<readCursorResponse>
  <items>
    <item>
      <id>450AB7C1.domain1.pol.100.16E3837.1.FD0.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-15T22:23:28Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>1</subject>
    </item>
    ...
  </items>
  <status>
    <code>0</code>
  </status>
</readCursorResponse>

```

## Stubbed Items

You can search on the statusExt field to find stubbed items. The values you can pass in the filter for the statusExt field are:

- ♦ stubbed (thirdParthStubbed)
- ♦ downloaded (thirdPartyDownloaded)
- ♦ archived (thirdPartyArchive)

The following example searches for stubbed items:



```

<getItemRequest>
  <container>4E2E7543.domain.P01.100.1646A30.1.44.1@13</container>
  <view>default stubbed</view>
  <filter>
    <element>
      <op>and</op>
      <element>
        <op>isOf</op>
        <field>statusExt</field>
        <value>stubbed</value>
      </element>
      <element>
        <op>exists</op>
        <field>statusExt</field>
        <value>0</value>
      </element>
    </element>
  </filter>
</getItemRequest>

```

---

**NOTE:** You must search for the existence of the statusExt field and the stubbed value being set or you get false positive matches.

---

## HTTP Connections

GroupWise Web Services is based on HTTP as the transport layer. HTTP 1.1 provides persistent connections to the POA. The `<connection>keep-alive</connection>` line establishes this persistent connection. If the connection is not used, it times out after 30 minutes of inactivity. Most development frameworks reestablish the connection for an application after the timeout. You should test your application to ensure that it behaves correctly after a timeout.

There is also a SOAP session connection. This session is a GroupWise connection to the POA. The session times out after 30 minutes. If your application needs to keep this GroupWise session alive, the `getTimestampRequest` method provides a `noop` parameter. The `noop` option keeps the session and HTTP connections open.

The following is an example of HTTP headers in a login request and response:

```

<HTTPHeaders>
  <content-type>text/xml; charset=utf-8</content-type>
  <content-length>809</content-length>
  <soapaction>"loginRequest"</soapaction>
  <user-agent>Java/1.5.0_04</user-agent>
  <host>137.65.15.37:8080</host>
  <accept>text/html, image/gif, image/jpeg, *; q=.2,
    */*; q=.2</accept>
  <connection>keep-alive</connection>
</HTTPHeaders>

<HTTPHeaders>
  <date>Tue, 10 Oct 2012 12:01:19 GMT</date>
  <server>Windows GroupWise POA 7.0.1</server>
  <content-type>text/xml; charset=utf-8</content-type>
  <content-length>940</content-length>
  <pragma>no-cache</pragma>
</HTTPHeaders>

```

## 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.

- ♦ [“Large Message Bodies” on page 34](#)
- ♦ [“HTML Message Bodies” on page 36](#)

## 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 returned 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.pol.100.1647535.1.635.1@1:7.
      domain1.pol.100.0.1.0.1@16</id>
    ...
    <subject>Large Message Body</subject>
    <message>
      <part contentType="text/plain" length="1008596" id="452A4D93.
        domain1.pol.200.20000B2.1.374.1@65:452A4D93.domain1.pol.100.
          1647535.1.635.1@1:7.domain1.pol.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 16 KB.

```

<getAttachmentRequest>
  <id>452A4D93.domain1.pol.200.20000B2.1.374.1@65:452A4D93.domain1.
    pol.100.1647535.1.635.1@1:7.domain1.pol.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.pol.200.20000B2.1.374.1@65:452A4D93.
    domain1.pol.100.1647535.1.635.1@1:7.domain1.pol.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.

## 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.pol.100.1647535.1.644.1@1:7.
    domain1.pol.100.0.1.0.1@16</id>
  ...
  <message>
    <part contentType="text/plain"
      length="284">elxydGYxXGRlZmYwelxmb250dGJselxmMFxmbmlsXGZj
      aGFyc2V0MSBUYWhvbWE7fXl7XGNvbG9ydGJsOlxyZWQyNTVcZ3JlZW4wX
      GJsdWUwOlxyZWQyNTVcZ3JlZW4wXGJsdWUwO3l7XHN0eWxlc2hlZXR7XGZz
      MTYgR3JvdXBXaXNlVmllldzt9fVxmczE2IEhUTUwgbWVzc2FnZSB3aXRoIGF
      uIGltYWdlLlxlwYXIgIFxwYXIgIFxwYXIgQWZ0ZXIgdGhlIHBPY3Rl
      cmUuXHBhcnB9</part>
    </message>
  <attachments>
    <attachment>
      <id>452A6054.domain1.pol.200.20000B2.1.398.1@45:452A6054.
        domain1.pol.100.1647535.1.644.1@1:7.domain1.pol.
        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.pol.200.20000B2.1.399.1@45:452A6054.
        domain1.pol.100.1647535.1.644.1@1:7.domain1.pol.
        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.pol.200.20000B2.1.398.1@45:452A6054.
    domain1.pol.100.1647535.1.644.1@1:7.domain1.pol.
    100.0.1.0.1@16</id>
</getAttachmentRequest>

<getAttachmentResponse>
  <part length="548"
    offset="409">PEhUTUw+PEhFQUQ+DQo8TUVUQSBodHRwLWVxdWl2PUN
    vbnRlbnQtVHlwZSB
    ...
    ERJVj5BZnRlciB0aGUgcGljdHVyZS48L0RJVj48L0JPRFk
    +PC9IVElMPg==
  </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>

```

## Proxy

Proxy allows users to share their accounts with other users. For example, user1 wants to provide access to user2 to his or her account. User1 needs to provide access rights to user2. The [createProxyAccessRequest](#) method grants access rights to user2, [modifyProxyAccessRequest](#) modifies the access rights, and [removeNotifyRequest](#) removes any previously defined rights.

User2 now wants to proxy into user1's account. The [getProxyListRequest](#) method returns the list of users that user2 has successfully proxied into in the past. If user2 has not successfully proxied into user1's account in the past, the user1 account does not appear in the [getProxyListResponse](#).

For user2 to proxy into user1's account, call the [loginRequest](#) method.

## Recurrence

GroupWise supports recurring appointments, notes, and tasks. Recurring items are created from one item that fans out into many items. For example, a manager wants to send out a team appointment. The appointment occurs every Monday morning at 9:00 a.m. The manager could send out one appointment for each occurrence, but that would be time-consuming and tedious. Using GroupWise, the manager can send out one appointment and specify the recurring dates for the appointment.

This section contains the following examples:

- ♦ [“Recurring Item Example” on page 39](#)
- ♦ [“Recurring Daily Appointment Example” on page 39](#)
- ♦ [“Recurring Item with a Rule Example” on page 40](#)

GroupWise Web Services defines recurring items on the `CalendarItem` element. The recurring elements are `rdate`, `exdate`, and `rrule`.

The `rdate` element specifies a list of days to include in the recurrence. The `rrule` specifies a rule that defines a list of days to include in the recurrence. The `exdate` specifies a list of days to exclude in the recurrence. All three elements (`rdate`, `rrule`, and `exdate`) can be used to build a list of recurring days.

The `rrule` element defines a rule that defines the number of occurrences for the calendar item. There are industry standards that deal with recurrence. You should be familiar with recurrence in RFC 2445. For more information, see [4.3.10 Recurrence Rule \(http://www.ietf.org/rfc/rfc2445.txt\)](http://www.ietf.org/rfc/rfc2445.txt).

GroupWise supports a subset of the many definitions that are available in recurrence rules. For example, in `rrule` there is a frequency element defined in the GroupWise schemas. RFC 2445 defines Secondly, Minutely, Hourly, Daily, Weekly, Monthly, and Yearly. At this time, GroupWise doesn't support Secondly, Minutely, or Hourly. Consult the GroupWise schemas and RFC 2445 to see what elements are supported.

For each recurring instance, GroupWise creates a new item in the user's database at send time. For example, suppose an application creates a recurring appointment by calling `sendItemRequest`. The appointment defines a recurrence rule. When the POA evaluates the item, it fans out or creates a new item for each instance defined in the recurrence rule. This fanning out of recurring items is different than some other collaboration products. Other collaboration products create one instance of the item and store the recurrence rule with the item.

When you get a list of calendar items, recurring items are grouped together with a recurrenceKey. If you want to accept, decline, delegate, or modify all recurring items, pass in the recurrenceKey to the appropriate method.

If your user base is used to seeing only one calendar item to represent a recurring calendar item, it is up to the application to collapse the fanned-out items into one item. The recurrenceKey is used to determine which items are a part of the recurrence group.

## Recurring Item Example

In the following example, three appointments are created in the recipient's mailbox by calling [sendItemRequest](#). The startDate and endDate create one instance, while the rdate element defines two other instances. The startDate and endDate both have a time associated with the date. This time value is used to create the duration of the appointment. The rdate values have only a date with no associated time. The rdate elements use the time and duration specified in the startDate and endDate elements.

```
<sendItemRequest>
  <types:item type="Appointment">
    ...
    <subject>Recurrence 1</subject>
    ...
    <startDate>20121012T180000Z</startDate>
    <endDate>20121012T190000Z</endDate>
    <rdate>
      <date>20121013</date>
      <date>20121016</date>
    </rdate>
    <alarm types:enabled="1">5</alarm>
  </types:item>
</sendItemRequest>
```

## Recurring Daily Appointment Example

In the following example, a recurring item is created on a startDate of 10/12/2012. The rrule element specifies to create an appointment daily until 10/20/2012.

```
<sendItemRequest>
  <types:item type="Appointment">
    <subject>Recurrence 2</subject>
    <startDate>20121012T180000Z</startDate>
    <endDate>20121012T190000Z</endDate>
    <rrule>
      <frequency>Daily</frequency>
      <until>20121020</until>
    </rrule>
    <alarm types:enabled="1">5</alarm>
  </types:item>
</sendItemRequest>
```

## Recurring Item with a Rule Example

In the following example, recurring items are created using this rule:

- ♦ startDate specifies October 12, 2012 at 7:00PM UTC as the start time.
- ♦ endDate specifies October 12, 2012 at 8:00PM UTC as the end time.
- ♦ frequency creates the appointments on a monthly basis.
- ♦ until creates the monthly appointments until 2/28/2013.
- ♦ day creates the appointments only on Mondays.

```
<startDate>20121012T190000Z</startDate>
<endDate>20121012T200000Z</endDate>
<rrule>
  <frequency>Monthly</frequency>
  <until>20070228</until>
  <interval>2</interval>
  <byDay>
    <day>Monday</day>
  </byDay>
</rrule>
```

## Redirection

Redirection is used to discover a user's post office address and port. If you try to log in as a user to a wrong POA, a `redirectToHost` element is returned in the `loginResponse`. The `redirectToHost` element provides the IP address or DNS name and port of the POA of the user. Your application can retry the `loginRequest` to the new IP address and port. It can also cache the correct IP address and port so that a redirection is not required in the future.

For redirection to work correctly, the SOAP port needs to be configured in ConsoleOne for each POA. ConsoleOne writes the SOAP port settings to the GroupWise databases. On a redirection attempt, other POAs can successfully read the settings from the databases. If a POA is configured in a configuration file or on the command line, other POAs cannot access configuration files or command line settings. The correct settings must be in the GroupWise databases.

In the following [loginRequest](#) example, user `po2u1` logged into the wrong POA. The `loginResponse` returned the correct IP address and port of the user. It also returned error 59923 with a description stating a redirection is required.

```
<loginRequest>
  <types:auth types:PlainText>
    <username>po2u1</username>
    <password>po2u1</password>
  </auth>
  <application>BEEPClient</application>
</loginRequest>
```



```

<loginResponse>
  <redirectToHost>
    <ipAddress>137.65.15.37</ipAddress>
    <port>7192</port>
  </redirectToHost>
  <status>
    <code>59923</code>
    <description>Redirect user to different PO.</description>
  </status>
</loginResponse>

```

## Streaming Attachments

Before using the streaming code, verify that you are logged into a GroupWise 8.0 or later POA. After a successful login, the loginResponse contains the POA version in the gwversion element.

The streaming functionality is accessible via the following URL:

*http(s)://server:port/attachment?session=...&id=...*

- ♦ http or https: HTTP protocol with or without SSL.
- ♦ server: The IP address of the DNS name of the POA server.
- ♦ attachment: The keyword to tell the POA to stream the attachment.
- ♦ session: A valid login session is required.
- ♦ id: The SOAP id of the attachment to be streamed
- ♦ port: The SOAP port.

There are some additional parameters that can be included in the URL:

- ♦ &plain=1: Returns the document in plain text.
- ♦ &mime=1: Returns the MIME representation (RFC822) of the item. You need to pass in the ID of the item and not the attachment when getting the MIME representation.

A GroupWise error in streaming the attachment is returned as HTTP status 400. The header line contains X-GWError-Code = GroupWise Error Code, which specifies the GroupWise error code.

Sample Java code is given below:

```

m_item = (Mail)res.getItem();
info = m_item.getAttachments();
items = info.getAttachment();
for ( i = 0; i < items.length; i++ )
{
    str = temp + m_gwMain.getGroupWiseServiceId() + "&id=" +
items[i].getId().get_value();
    size = (int)items[ i ].getSize();
    offset = 0;
    try
    {
        URL url = new URL( str );
        HttpURLConnection huc = (HttpURLConnection)url.openConnection();
        InputStream is = huc.getInputStream();

```

```

        byte[] b = new byte[ 16384 ];
        out = new FileOutputStream( new File( "c:\\temp", items[ i ].getName()
    ) );
        System.out.println( "Start: " + new Date().toString() );
        while ( size > 0 )
        {
            len = is.read( b, 0, 16384 );
            if ( -1 == len ) {
                break;
            }
            out.write( b, 0, len );
            offset += len;
            size -= len;
        }
        out.flush();
        out.close();
        out = null;
        System.out.println( "End: " + new Date().toString() );
    }
    catch ( Exception ex )
    {
        ex.printStackTrace();
        break;
    }
}

```

In GroupWise 8.0 SP2 and later, the ability to stream attachments into GroupWise using HTTP PUT was also added. The format of the URL is similar:

`http(s)://server:port/attachment?session=...&id=...&name=...`

- ♦ `http` or `https`: HTTP protocol with or without SSL.
- ♦ `server`: The IP address of the DNS name of the POA server.
- ♦ `attachment`: The keyword to tell the POA to stream the attachment.
- ♦ `session`: A valid login session is required.
- ♦ `id`: The SOAP id of the item to receive the streamed attachment.
- ♦ `port`: The SOAP port.
- ♦ `name`: The name of the attachment.

A personal attachment can be added to any item. If you want to send an item and stream the attachments to the item, it is a multi-step process. You first create a draft item. You then stream the attachments to the draft item. You then send an item using the `sendItemRequest`. You supply a link element in the item in the `sendItemRequest` referencing the draft item. The values are first taken from the item passed in the `sendItemRequest`. The information from the linked item is then added to the item (including the attachments). After the item is sent, the draft item is purged.

The following Java code shows how to send a large attachment using the streaming code:

```

public void testStreamAttachment() {
    byte[] bytes;
    byte[] b = new byte[ 16384 ];
    Calendar end;
    Calendar start;
    Distribution dist = new Distribution();
    File file;
    FileInputStream in;
    HttpURLConnection huc;
    int code;
    int len;
    LinkInfo link = new LinkInfo();
    Mail mail = new Mail();
    MessageBody mb = new MessageBody();
    MessagePart[] mp = new MessagePart[1];
    OutputStream os;
    Recipient[] recip = new Recipient[1];
    RecipientList list = new RecipientList();
    SendItemResponse resp = null;
    String id;
    String str;
    URL url;

    try {
        // time the send
        start = Calendar.getInstance();

        // setup a draft item
        mail.setSubject( "Test of HTTP PUT" );
        mail.setSource( ItemSource.draft );

        // get the message body
        bytes = "Message Body!".getBytes( "UTF-8" );
        mp[0] = new MessagePart();
        mp[0].set_value( bytes );
        mb.setPart( mp );
        mail.setMessage( mb );

        // set the recipient
        recip[ 0 ] = new Recipient();
        recip[ 0 ].setDisplayName( "Preston Stephenson" );
        recip[ 0 ].setEmail( "pstephenson@prestons.provo.novell.com" );
        recip[ 0 ].setDistType( DistributionType.TO );
        list.setRecipient( recip );
        dist.setRecipients( list );
        mail.setDistribution( dist );

        // create the draft item
        resp = m_main.getService().sendItemRequest( mail,
m_main.getSessionId(),
            m_main.getTrace() );
        if ( 0 != resp.getStatus().getCode() ) {
            m_main.displayError( resp.getStatus(), "testStreamAttachment" );
        }
    }
}

```

```

// get the draft item id, encode it for the URL
id = URLEncoder.encode(resp.getId()[0], "UTF-8");

// set up the URL
str = "http://" + m_main.getLogin().getServer() + ":" +
      m_main.getLogin().getPort() + "/attachment?session=" +
      m_main.getSessionId() + "&id=" + id + "&name=onemb.txt";
url = new URL( str );

// set up the file to stream
file = new File( "c:\\temp", "onemb.txt" );
str = String.valueOf( file.length() );
huc = (URLConnection)url.openConnection();
huc.setDoOutput( true );
huc.setRequestMethod( "PUT" );
huc.setFixedLengthStreamingMode( (int)file.length() );
os = huc.getOutputStream();
in = new FileInputStream( file );

// stream the attachment
for ( ;; ) {
    len = in.read(b);
    if ( -1 == len ) {
        break;
    }
    os.write( b, 0, len );
}
os.close();
code = huc.getResponseCode();
if ( code < 200 || code > 299 ) {
    m_main.displayError( resp.getStatus(), "testStreamAttachment" );
}

// set up the mail item to send
mail = new Mail();
recip[ 0 ] = new Recipient();
recip[ 0 ].setDisplayName( "Preston Stephenson" );
recip[ 0 ].setEmail( "pstephenson@prestons.provo.novell.com" );
recip[ 0 ].setDistType( DistributionType.TO );
list.setRecipient( recip );
dist.setRecipients( list );
mail.setDistribution( dist );

// set the link to the draft item
link.setType( LinkType.draft );
link.setId( resp.getId()[0] );
mail.setLink( link );

```

```

        // send the item
        resp = m_main.getService().sendItemRequest( mail,
m_main.getSessionId(),
            m_main.getTrace() );

        // get the elapsed time
        end = Calendar.getInstance();
        str = "Streaming: " + String.valueOf( end.getTimeInMillis() -
            start.getTimeInMillis() );
        m_main.displayError( resp.getStatus(), str );
    } catch ( Exception e ) {
        e.printStackTrace();
    }
}

```

---

**NOTE:** There is not much of a performance increase for streaming small attachments. You should consider streaming attachments over 64 KB in size.

---

## Trusted Application

Trusted applications allow third-party applications access to all accounts on a GroupWise system, without needing to know user passwords. In order for an application to use trusted application authentication, GroupWise administrators must create a trusted application name and key. The GroupWise administrator has control over the trusted application and decides which applications can log in to the GroupWise system using the trusted application name and key.

Call the [loginRequest](#) method to log in as a trusted application.

For more information on trusted applications, see the [NDK Trusted Application page \(https://www.novell.com/developer/ndk/groupwise\\_trusted\\_application\\_api.html\)](https://www.novell.com/developer/ndk/groupwise_trusted_application_api.html).

## Filters and Views

Filters help refine the search for items on a user's account. Views reduce the number of elements returned on each item.

The following sections contain more information:

- ♦ [“Filters” on page 46](#)
- ♦ [“Filtering on Different Item Types” on page 48](#)
- ♦ [“Views” on page 57](#)

## Filters

Filters refine a search for items. [createCursorRequest](#) and [getItemsRequest](#) use filters. Filters are defined in the `types.xsd` schema. You should keep your filters as simple as possible.

To create a simple filter, instantiate a `Filter` element and a `FilterEntry` element. Then, in the `FilterEntry` element, add a field, value, and operation. For example, a field could have a subject value that is set to "Novell." The operation could have a contains value. The filter then returns a list of items if the subject contains "Novell."

```
Filter gwFilter = new Filter();// Instantiate a Filter
FilterEntry gwFE = new FilterEntry();// Instantiate a FilterEntry

gwFE.setField("subject");
gwFE.setValue("Novell");
gwFE.setOp(FilterOp.contains);
gwFilter.setElement(gwFE);// getItemsRequest is called with the filter.
```

The `getItemsRequest` and Response follows:

```
<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>subject</view>
  <filter>
    <element type="FilterEntry">
      <op>contains</op>
      <field>subject</field>
      <value>Novell</value>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item>
      <id>4513CC01.domain1.pol.100.16E3837.1.1057.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>Novell announces Linux</subject>
    </item>
    <item>
      <id>4513CD2E.domain1.pol.100.16E3837.1.1059.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>Novell exceeds Sales</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>
```

To create a more complicated filter, instantiate a Filter element and two FilterEntry elements. Next, in both FilterEntry elements, add a field, value, and operation. Group the two FilterEntry elements with a FilterGroup element. The FilterGroup element defines a FilterOp with a limited set of values: and, or, and not.

The first FilterEntry returns items that have a subject that contains the string "Novell." The second FilterEntry returns items that have a date greater than a specific date. The two FilterEntry elements are grouped together with an and operation.

```
Filter gwFilter = new Filter();// Instantiate a Filter
FilterGroup gwFG = new FilterGroup();// Instantiate a FilterGroup
FilterEntry[] gwFE = new FilterEntry[2];// Instantiate two FilterEntries

gwFE[0] = new FilterEntry();// Fill in FilterEntry one
gwFE[0].setField("subject");
gwFE[0].setValue("Novell");
gwFE[0].setOp(FilterOp.contains);

gwFE[1] = new FilterEntry();// Fill in FilterEntry two
gwFE[1].setField("created");
gwFE[1].setValue("2012-09-22T17:45:00Z");
gwFE[1].setOp(FilterOp.gt);

gwFG.setOp(FilterOp.and);// Group the two FilterEntries with an "and"
gwFG.setElement(gwFE);// Add the two FilterEntries to the FilterGroup
gwFilter.setElement(gwFG);// Add the FilterGroup to the Filter
```

The getItemRequest and Response are shown below:

```
<getItemRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>subject</view>
  <filter>
    <element type="FilterGroup">
      <op>and</op>
      <element type="FilterEntry">
        <op>contains</op>
        <field>subject</field>
        <value>Novell</value>
      </element>
      <element type="FilterEntry">
        <op>gt</op>
        <field>created</field>
        <value>2012-09-22T17:45:00Z</value>
      </element>
    </element>
  </filter>
  <count>-1</count>
</getItemRequest>
```

```

<getItemsResponse>
  <items>
    <item>
      <id>4513CD2E.domain1.pol.100.16E3837.1.1059.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>Novell exceeds Sales</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>

```

## Filtering on Different Item Types

Filtering is based on the different item types: mail, the GroupWise Address Book, and personal address books.

You can search on dates relative to a floating date. The date options that are available are Today, Tomorrow, ThisMonth, ThisWeek, ThisYear, and Yesterday. These values are defined in the FilterDate element.

- ♦ [“Mail, Calendar, and Documents” on page 48](#)
- ♦ [“GroupWise Address Book” on page 53](#)
- ♦ [“Personal Address Book” on page 54](#)
- ♦ [“Attachments” on page 56](#)

## Mail, Calendar, and Documents

Filters for mail, calendar, and documents use the following schema element names for the FilterEntry field. Some fields accept only certain FilterOp operations.

- ♦ created
- ♦ delivered
- ♦ modified
- ♦ startDate
- ♦ startDay
- ♦ endDate
- ♦ endDay
- ♦ dueDate
- ♦ @type—use FilterOp.eq on a single type, or use FilterOp.IsOf on multiple item types
- ♦ to, bc, cc—string fields that are not a part of the recipient list
- ♦ subject
- ♦ source—use only FilterOp.isOf or FilterOp.isNotOf or FilterOp.eq



- ♦ status—use only FilterOp.isOf or FilterOp.isNotOf or FilterOp.eq
- ♦ msgId
- ♦ displayName
- ♦ from/displayName
- ♦ creator/displayName
- ♦ author/displayName
- ♦ versionCreator/displayName
- ♦ email
- ♦ from/email
- ♦ creator/email
- ♦ author/email
- ♦ versionCreator/email
- ♦ uuid
- ♦ from/uuid
- ♦ creator/uuid
- ♦ author/uuid
- ♦ versionCreator—use FilterOp.eq only
- ♦ iCalId
- ♦ text—a QuickFinder search that searches all text fields, message body, and attachments
- ♦ acceptLevel—use only FilterOp.eq
- ♦ place
- ♦ subType—used with custom messages
- ♦ caller, company, phone (Phone messages)
- ♦ versionDescription
- ♦ size (attachment size)
- ♦ custom field or user-defined field
- ♦ allDayEvent (**--for GroupWise 2012 and later**)
- ♦ originalId (**--for GroupWise 2012 and later**)
- ♦ documentNumber (**--for GroupWise 2012 and later**)
- ♦ versionNumber (**--for GroupWise 2012 and later**)
- ♦ category (**--for GroupWise 2012 and later**)
- ♦ originalSubject (**--for GroupWise 2012 and later**)
- ♦ taskCategory (**--for GroupWise 2012 and later**)
- ♦ taskPriority (**--for GroupWise 2012 and later**)
- ♦ xField (**--for GroupWise 2012 and later**)
- ♦ retentionModified (**--for GroupWise 2012 and later**)
- ♦ folder (**--for GroupWise 2012 and later**)

- ♦ sid (**--for GroupWise 2012 and later**)
- ♦ assignedDate (**--for GroupWise 2012 and later**)

This section contains the following examples:

- ♦ [“Mailbox Example” on page 50](#)
- ♦ [“Sent Items Example” on page 50](#)
- ♦ [“Searching Elements Example” on page 51](#)
- ♦ [“Relative Example” on page 52](#)

## Mailbox Example

Suppose you want to return only appointments in the Mailbox by calling [getItemsRequest](#). Use the @type attribute in the field value. The @ signals an attribute in the schema.

The following example request and response use the @type = Appointments on the Mailbox container:

```
<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>subject</view>
  <filter>
    <element type="FilterEntry">
      <op>eq</op>
      <field>@type</field>
      <value>Appointment</value>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item type="Appointment">
      <id>44FEDC3B.domain1.pol.100.1676834.1.76F.1@4:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>Delegated Appt</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>
```

## Sent Items Example

The following example returns all items from a container that are sent items by calling [getItemsRequest](#). The value in the filter could be replaced with received, sent, draft, or personal.

```

<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>subject</view>
  <filter>
    <element type="FilterEntry">
      <op>eq</op>
      <field>source</field>
      <value>sent</value>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item>
      <id>44E0A6E4.domain1.pol.100.16E3837.1.EFD.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>sent</source>
      <subject>moving an item</subject>
    </item>
    ...
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>

```

## Searching Elements Example

The following example searches all text element fields, including the message body and attachments for a specific string by calling [getItemsRequest](#):

```

<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>subject message</view>
  <filter>
    <element type="FilterEntry">
      <op>eq</op>
      <field>text</field>
      <value>Test of the emergency broadcast system.</value>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>

```

```

<getItemsResponse>
  <items>
    <item>
      <id>4513F160.domain1.pol.100.16E3837.1.105B.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>Test mail message</subject>
      <message>
        <part contentType="text/plain"
          length="112">VGhpcyBpcyBhIHRlc3Qgb2YgdGhlIGVtZXJnZW5
            jeSBicm9hZGNhc3Qgc3lzdGVtLiAgVGhpcyBpcyBvbmx5IGEgdGV
            zdC4KCmFzZGZhc2RmCg==
          </part>
        </message>
      </item>
    </items>
    <status>
      <code>0</code>
    </status>
  </getItemsResponse>

```

## Relative Example

The following example demonstrates retrieving all items with a startDate greater than or equal to two days before today by calling [getItemsRequest](#).

In the date element below, the relative date is Today. The field is the startDate with a value of -2. Using -2 as the value subtracts two days from Today.

```

<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>subject</view>
  <filter>
    <element type="FilterEntry">
      <op>fieldGTE</op>
      <field>startDate</field>
      <value>-2</value>
      <date>Today</date>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>

```

```

<getItemsResponse>
  <items>
    <item type="Mail">
      <id>452A40F1.domain1.pol.100.16E3837.1.109B.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>personal</source>
      <subject>Relative Date Example</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>

```

## GroupWise Address Book

Creating a filter for GroupWise Address Book items can use the following schema element names for the FilterEntry field. Some fields take only certain FilterOp operations. The email address is not a part of the fields that are available to search. The email address is not indexed. It is built from the parts, such as displayName@iDomain.

- ♦ created
- ♦ @type—use FilterOp.eq only
- ♦ department
- ♦ fullName/firstName
- ♦ fullName/lastName
- ♦ username—login ID to GroupWise
- ♦ uuid—use FilterOp.eq only
- ♦ domain
- ♦ postOffice

Filtering the GroupWise Address Book using firstName, lastName, displayName, or department is fastest because the fields are indexed. If you filter using any of the other fields, the searches are time-consuming because the fields are not in the index.

The following example demonstrates calling [getItemsRequest](#):

```

<getItemsRequest>
  <container>GroupWiseSystemAddressBook@52</container>
  <view>null</view>
  <filter>
    <element type="FilterEntry">
      <op>eq</op>
      <field>lastName</field>
      <value>LAST01010000</value>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item type="Contact">
      <id>07B4A660-18CA-0000-B989-
        BA00EB000D00@56:GroupWiseSystemAddressBook@52</id>
      <name>FIRST01010000 LAST01010000</name>
      <version>1</version>
      <modified>2005-10-31T20:06:55Z</modified>
      <container>GroupWiseSystemAddressBook@52</container>
      <created>2005-10-31T20:06:39Z</created>
      <uuid>07B4A660-18CA-0000-B989-BA00EB000D00</uuid>
      <domain>domain1</domain>
      <postOffice>po2</postOffice>
      <distinguishedName>USER01010000.SERVER</distinguishedName>
      <userid>USER01010000</userid>
      <fullName>
        <displayName>FIRST01010000 LAST01010000</displayName>
        <firstName>FIRST01010000</firstName>
        <lastName>LAST01010000</lastName>
      </fullName>
      <emailList primary="USER01010000@phantom.com">
        <email>USER01010000@phantom.com</email>
      </emailList>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>

```

## Personal Address Book

Creating a filter for personal address book items can use the following schema element names for the FilterEntry field. Some fields only take certain FilterOp operations.

- ♦ @type—use FilterOp.eq only
- ♦ department
- ♦ displayName or name
- ♦ fullName/firstName
- ♦ fullName/lastName

- ♦ fullName/middleName
- ♦ fullName/namePrefix
- ♦ fullName/nameSuffix
- ♦ postOffice
- ♦ username—login id to GroupWise
- ♦ uuid—use FilterOp.eq only
- ♦ emailList/email
- ♦ emailList/@primary
- ♦ created
- ♦ modified
- ♦ organization (for GroupWise 2012 and later)

The following example returns group items in a personal address book by calling [getItemsRequest](#):

```
<getItemsRequest>
  <container>42C510EA.domain1.pol.104.16E3837.1.3.1@53</container>
  <view>null</view>
  <filter>
    <element type="FilterEntry">
      <op>eq</op>
      <field>@type</field>
      <value>Group</value>
    </element>
  </filter>
  <count>-1</count>
</getItemsRequest>
```

```
<getItemsResponse>
  <items>
    <item type="Group">
      <id>44FD7AC1.domain1.pol.104.16E3837.1.FF.1@58:42C510EA.
        domain1.pol.104.16E3837.1.3.1@53</id>
      <name>G2</name>
      <version>1</version>
      <modified>2012-09-05T19:25:21Z</modified>
      <container>42C510EA.domain1.pol.104.
        16E3837.1.3.1@53</container>
      <created>2012-09-05T19:25:21Z</created>
      <comment>asdf</comment>
    </item>
    <item type="Group">
```

```

        <id>44EB3142.domain1.pol.104.16E3837.1.F0.1@58:42C510EA.
            domain1.pol.104.16E3837.1.3.1@53</id>
        <name>Group</name>
        <version>5</version>
        <modified>2012-09-05T15:12:09Z</modified>
        <container>42C510EA.domain1.pol.104.16E3837.
            1.3.1@53</container>
        <created>2012-08-22T22:30:58Z</created>
    </item>
</items>
<status>
    <code>0</code>
</status>
</getItemsResponse>

```

## Attachments

It might be useful to search for items that have a certain type of attachment.

- ♦ SharedAttachmentTypes - (shared attachments)
- ♦ PersonalAttachmentTypes - (personal attachments)

The filter operations that can be used with these fields are:

- ♦ exists
- ♦ isOf
- ♦ isNotOf

The values that can be in the value element are:

- ♦ MessageBody
- ♦ File
- ♦ Recipients

The value string is a space delimited combination of the above values.

You have to test for the existence of the field as well as the values. For example, to return items that have a personal message body or HTML message body (text.htm), the filter would be:

```

<filter>
  <element xsi:type="FilterGroup">
    <op>and</op>
    <element xsi:type="FilterEntry">
      <op>exists</op>
      <field>PersonalAttachmentTypes</field>
    </element>
    <element xsi:type="FilterEntry">
      <op>isOf</op>
      <field>PersonalAttachmentTypes</field>
      <value>MessageBody File</value>
    </element>
  </element>
</filter>

```



Using the `isNotOf` operation is tricky. Contact development if you want to use this operation. Here is an example of getting items that have shared attachments only (no personal):

```
<filter>
  <element xsi:type="FilterEntry">
    <op>notExist</op>
    <field>PersonalAttachmentTypes</field>
  </element>
</filter>
```

## Views

This section contains the following topics:

- ♦ [“View Elements” on page 61](#)
- ♦ [“View Examples” on page 63](#)

Views reduce the elements returned on each item. Your applications can leverage views to retrieve only the item elements you are interested in. For example, suppose an application is retrieving items in the mailbox folder. You need to display only the subject and modified date. The application can specify a subject modified view. Each item in turn returns the subject and modified date. There are a few additional fields that are returned, such as the container, the item ID, and the source.

The view element is a space-delimited list of leaf elements defined in `types.xsd` schema. For example, `types.xsd` defines the subject element as `<subject " type="string"/>`. The element name is `subject`, which can be added to the view.

The following methods have a view parameter:

- ♦ [createCursorRequest \(page 88\)](#)
- ♦ [forwardRequest \(page 104\)](#)
- ♦ [getDeltasRequest \(page 128\)](#)
- ♦ [getFolderRequest \(page 136\)](#)
- ♦ [getFolderListRequest \(page 138\)](#)
- ♦ [getItemRequest \(page 146\)](#)
- ♦ [getItemsRequest \(page 148\)](#)
- ♦ [getQuickMessagesRequest \(page 169\)](#)
- ♦ [replyRequest \(page 240\)](#)

It is a good idea to look at the responses for the respective requests to determine the type of elements that can be added to the view. It does not make sense to add `subject` to the view when calling `getFolderListRequest` because `getFolderListRequest` does not return a `subject`.

Some important view strings are as follows:

### **all**

The administrator can limit what fields are returned when getting the GroupWise Address Book entries. If you supply this value and you have logged in as a trusted application, you can get back all of the fields.

This is also used to get all the status from the access right list in a shared folder.

**assignedDate**

When a task is originally created, the startDate is also put in the assignedDate field. After the dueDate is reached and the task has not yet been completed, the startDate is changed to the current date.

For GroupWise 2012 and later.

**attachment or attachments**

The attachment reference is returned with the item. getAttachmentRequest is used to retrieve the actual file attachments.

The HTML message body is included as an attachment. It is the first attachment and is named `text.htm`.

Retrieve the attachment information only when necessary. This view requires the POA to do more work, which affects performance.

**checklist**

Returns the checklist/tasklist elements: sequence number, due date, percent completed, completed date, and checklist thread.

**clientMessageld**

Returns the clientMessageld. This is the same id that you will see when you open an item in the GroupWise Windows client. Select Properties and find the Message Id.

**contactNotes**

Starting in GroupWise 8.0 SP3, contact notes are not returned by default when getting contacts from a personal address book. You must now pass this value to get the contact notes.

For GroupWise 2012 and later.

**count**

If the item type is a folder, a count of items that are in the folder is returned.

If the item type is a Contact (UserContact) folder or personal address book, the count of items in the container is returned.

For GroupWise 2012 and later.

**custom**

To get a specific custom value returned, you must specify the custom field name prefixed with "custom/". For example, if you want to get the custom field "MyAppRead", you would put "custom/MyAppRead" in the view. You must list each custom field name separately. Here is an example of getting two different custom fields:

```
<view>default custom/MyAppRead custom/alexStrField</view>
```

**default**

An important subset of elements is returned, based on the item type. Additional keywords can be included with the default keyword, which reduces the need to include all elements in the view.

The default keyword returns a different set of elements, based on the item type.

## flush

The user's folder list is cached while the user is logged in. Use the flush keyword to make sure a fresh copy of the folders and their attributes are read when accessing a folder.

## format

The subject text can include certain formatting codes (underline, bold and italics). To get these formatting codes returned, specify "format" in the view. The allowed formatting codes in the text are:

```
<b>    bold on
</b>   bold off
<i>    italics on
</i>   italics off
<u>    underline on
</u>   underline off
```

These values are only on the subject field or the originalSubject field if the subject is personalized. Here is an example of a personalized subject::

```
<subject>This is my subject!</subject>
<originalSubject>&lt;b&gt;This&lt;/b&gt; &lt;u&gt;is&lt;/u&gt;
    the &lt;i&gt;subject&lt;/i&gt;!</originalSubject>
```

Note: the '<' and '>' are escaped in the XML. The strings would be:

```
"This is my subject!"
"<b>This</b> <u>is</u> the <i>subject</i>!"
```

## hidden

Hidden items are returned.

## members

Returns the members of a personal address group.

## message or message/RTF

The message body is returned. If message is included, the plain text message body is returned. If message/RTF is included, the rich text format of the message body is returned.

---

**NOTE:** The plain text message body can be stored in two different formats, plain text or RTF. If you want to get the message body as plain text (no RTF codes), you request "message" in the view. If you want to get back the message body, with the RTF codes, you request "message/RTF". If the message body is stored as plain text (no RTF codes), the message body will always be returned as plain text. We do not add RTF codes to the plain text.

---

Retrieve the message body only when necessary. This view requires the POA to do more work, which affects performance.

## messageID

A shortened version of the item ID is returned. This shortened version matches the IMAP item ID. The SOAP version of the item ID is msgID.

**noContactNotes**

In GroupWise 8.0 SP2, it was determined that getting the contact notes on contacts could significantly affect the performance of the POA. This keyword was added to not get the contact notes.

For GroupWise 8.0 SP2 and later.

**noDisplay**

In GroupWise 8.0, the display settings for folders were returned by default. Setting noDisplay prevents those settings from being returned.

For GroupWise 8.0 SP2 and later.

**noDownload**

Usually, when the peek flag is passed in the view and the message body is requested, the item is not marked as opened. A third-party downloaded flag is set to indicate that the user may have seen the message body. In some cases, this is undesirable. It gives a user an indication that someone or some application has viewed the message body. Pass this keyword to prevent the third-party downloaded flag from being set. This flag is dependent on the application having logged into the user's account with a trusted application name and key.

For GroupWise 8.0 SP3 and later.

**normal**

By default, if you log in with a trusted application, you get all entries from the GroupWise Address Book (even the hidden ones). If you want to be able to only get the entries the user has rights to see, you must pass "normal" in the view.

**peek**

If an application reads an item and its message body, the item is marked as opened and read. With the peek keyword, however, the item is marked as opened but not read. The item is marked as downloaded by a third party.

**recipient or recipients**

The recipients list is returned. Typically, the recipient or recipients keyword is required only if you are responding to a distribution list. If you need to display the recipients, there is a string representation of the To, CC, and BC fields that are returned with the default keyword.

Retrieve the recipient's information only when necessary. This view requires the POA to do more work, which affects performance.

**recipientStatus**

When a user sends an item to other users, recipientStatus information is tracked. On sent items, the recipientStatus information is returned. Also, recipientStatus is available to the received item if you are the sender of the item. The recipientStatus implies recipients and recipient information is returned also.

Retrieve the recipientStatus information only when necessary. This view requires the POA to do more work, which affects performance.

**rssURL**

Returns the RSS URL.

**sid**

If you put sid in the view, the <sid> values are returned as the values in the <item> elements.  
For GroupWise 8.0 HP1 and later.

**sids**

If you put sid and sids in the view, the <sid> values are returned in a space delimited <sids> element instead of the <items> element.  
For GroupWise 2012 and later.

**stubbed**

Returns stubbed items. See the stubbing/archive documentation for more information.

**threading**

Returns the discussion threading for an item.

**unreadCount**

If the item type is a folder, the count of unread items that are in the folder is returned.

**xfield**

Returns the MIME RFC 822 X-Field values.  
For GroupWise 2012 and later.

## View Elements

The following list of elements can be added to the view:

- ♦ message
- ♦ message/RTF
- ♦ acceptLevel
- ♦ allDayEvent
- ♦ itemType
- ♦ startDate
- ♦ delivered
- ♦ source
- ♦ company
- ♦ caller
- ♦ phone
- ♦ created
- ♦ endDate
- ♦ to
- ♦ cc
- ♦ bc
- ♦ from
- ♦ email

- ♦ name
- ♦ timezone
- ♦ attachment
- ♦ attachments
- ♦ peek
- ♦ default
- ♦ hidden
- ♦ modified
- ♦ id
- ♦ msgId
- ♦ messageId
- ♦ place
- ♦ subject
- ♦ priority
- ♦ recipientStatus
- ♦ uuid
- ♦ version
- ♦ security
- ♦ statusTracking
- ♦ requestReply
- ♦ sendoptions
- ♦ iCalId
- ♦ recurrenceKey
- ♦ recipients
- ♦ recipient
- ♦ recipientStatus
- ♦ hasAttachments
- ♦ count
- ♦ unreadCount
- ♦ parent
- ♦ description
- ♦ sequence
- ♦ all
- ♦ container
- ♦ priority
- ♦ expires
- ♦ created
- ♦ size

- ♦ smimeType
- ♦ subType
- ♦ mimeEncoding
- ♦ nntpOrImap
- ♦ alarm
- ♦ all (for GroupWise Address Book fields)
- ♦ category
- ♦ checklist
- ♦ clientMessageld
- ♦ flush
- ♦ members
- ♦ normal (for GroupWise Address Book entries)
- ♦ originalSubject
- ♦ rssURL
- ♦ stored (Return the stored name folder name for a system folder instead of the language dependent name.)
- ♦ stubbed
- ♦ threading

The following list of elements were added in GroupWise 8.0 SP2:

- ♦ noDisplay
- ♦ noContactNotes

The following list of elements were added in GroupWise 8.0 SP3:

- ♦ contactNotes
- ♦ noDownload
- ♦ retentionModified

The following list of elements were added in GroupWise 2012:

- ♦ assignedDate
- ♦ pabName (Return the address book name as the name of the corresponding user contact folder.)
- ♦ xField

## View Examples

This section contains the following examples:

- ♦ [“Default View Example” on page 64](#)
- ♦ [“Expanded View Example” on page 65](#)

## Default View Example

The following example gets an item from the Mailbox with the default view by calling [getItemsRequest](#):

```
<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <view>default</view>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item>
      <id>43BA73CB.domain1.pol.100.1676834.1.6FB.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <version>3</version>
      <modified>2012-01-03T19:59:40Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <created>2012-01-03T19:53:31Z</created>
      <status>
        <opened>1</opened>
        <read>1</read>
      </status>
      <msgId>43BA73CB.domain1.pol.200.20000CB.1.111D.1</msgId>
      <source>received</source>
      <delivered>2012-01-03T19:53:31Z</delivered>
      <security>Normal</security>
      <subject>Fwd: Test Message from U1</subject>
      <distribution>
        <from>
          <displayName>ul</displayName>
          <email>ul.pol.domain1</email>
          <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
        </from>
        <to>u2</to>
        <sendoptions>
          <statusTracking>None</statusTracking>
        </sendoptions>
      </distribution>
      <options>
        <priority>Standard</priority>
      </options>
      <hasAttachment>1</hasAttachment>
      <size>1189</size>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>
```



## Expanded View Example

The following example is the same as the preceding one, except that it includes the attachments, recipients, message, and peek keywords by calling `getItemRequest`. It also contains the attachments, recipients, and message elements.

```
<getItemRequest>
  <id>4512B362.domain1.pol.100.1676834.1.798.1@1:7.
    domain1.pol.100.0.1.0.1@16</id>
  <view>default message attachments recipients peek</view>
</getItemRequest>

<getItemResponse>
  <item>
    <id>4512B362.domain1.pol.100.1676834.1.798.1@1:7.
      domain1.pol.100.0.1.0.1@16</id>
    <version>2</version>
    <modified>2012-09-21T21:44:48Z</modified>
    <container>7.domain1.pol.100.0.1.0.1@16</container>
    <created>2012-09-21T21:44:34Z</created>
    <status/>
    <msgId>4512B362.domain1.pol.200.20000CA.1.673.1</msgId>
    <messageId><4512B362.261E.00CA.0@phantom.com></messageId>
    <source>received</source>
    <delivered>2012-09-21T21:44:34Z</delivered>
    <security>Normal</security>
    <subject>Test message</subject>
    <distribution>
      <from>
        <displayName>u2</displayName>
        <email>u2@phantom.com</email>
        <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
      </from>
      <to>u2</to>
      <recipients>
        <recipient>
          <displayName>u2</displayName>
          <email>u2@phantom.com</email>
          <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
          <distType>TO</distType>
          <recipType>User</recipType>
        </recipient>
      </recipients>
      <sendoptions>
        <statusTracking>None</statusTracking>
      </sendoptions>
    </distribution>
    <message>
      <part contentType="text/plain"
        length="40">VGhpcyBpcyBhIGllc3NhZ2UgYm9keS4KCmFzZGYK</part>
    </message>
    <attachments>
      <attachment>
        <id>4512B362.domain1.pol.200.20000CA.1.671.1@45:
```

```

4512B362.domain1.pol.100.1676834.1.798.1@1:7.
domain1.pol.100.0.1.0.1@16</id>
<name>Web Services Abstracts.odt</name>
<contentType>application/vnd.oasis.
    opendocument.text</contentType>
<size>15500</size>
<date>2012-09-15T12:39:06Z</date>
<hidden>0</hidden>
</attachment>
</attachments>
<options>
    <priority>Standard</priority>
</options>
<hasAttachment>1</hasAttachment>
<size>16346</size>
</item>
<status>
    <code>0</code>
</status>
</getItemResponse>

```

## ContactNotes on a Contact

ContactNotes provide the ability to add a list of notes to a contact. These notes operate as attachments on a contact. Each attachment has a unique id and creation date. Comments are also available and have been augmented with ContactNotes.

Getting contact notes can place a heavy load on the POA. In 8.0 SP2, the keyword (“noContactNotes”) was added to not get the contact notes when getting the personal address book items.

In GroupWise 8.0 SP3, it was decided to make the default be that contact notes are not returned. You must explicitly ask for the contact notes by putting “contactNotes” in the view.

## Creation of a ContactNote

Below is a trace creating a ContactNote on an existing contact. The method modifyItemRequest is called with the id of the contact to modify.

```
<ns0:modifyItemRequest>
<ns0:id>48FEE952.domain1.pol.104.16A6163.1.4A.1@56:4847C63F.domain1.pol.10
4.16A6163.1.3.1@53</ns0:id>
<ns0:updates>
<ans1:add xmlns:ans1="http://schemas.novell.com/2005/01/GroupWise/types"
xsi:type="ans1:Contact">
<ans1:version>0</ans1:version>
<ans1:contactNotes>
<ans1:contactNote>
<ans1:attachment>
<ans1:contentType>text/plain</ans1:contentType>
<ans1:size>0</ans1:size>
<ans1:data>QWRkaW5nlHZpYSBTT0FQIC0gMQ==</ans1:data>
</ans1:attachment>
</ans1:contactNote>
</ans1:contactNotes>
</ans1:add>
</ns0:updates>
<ns0:recurrenceAllInstances>0</ns0:recurrenceAllInstances>
</ns0:modifyItemRequest>
```

Below is Java code that demonstrates how to create a ContactNote:

```
private void addContactNote()
{
    ModifyItemResponse mRes = new ModifyItemResponse();
    ItemChanges iChanges = new ItemChanges();

    ContactNote[] gwContactNote = new ContactNote[1];
    ContactNotes gwContactNotes = new ContactNotes();
    Contact gwContact = new Contact();
    AttachmentItemInfo attachment = new AttachmentItemInfo();
    String sNote = new String("Adding via SOAP - 1");
    byte[] gwByte = new byte[sNote.length()+1];

    try
    {
        gwByte=new byte[sNote.length()+1];
        gwByte=sNote.getBytes("UTF8");
    }
    catch (UnsupportedEncodingException ex)
    {
        ex.printStackTrace();
    }

    attachment=new AttachmentItemInfo();
    attachment.setContentType("text/plain");
    attachment.setData(gwByte);

    gwContactNote[0]=new ContactNote();
```

```

        gwContactNote[0].setAttachment(attachment);
        gwContactNote.setContactNote(gwContactNote);
        gwContact.setContactNotes(gwContactNotes);
        iChanges.setAdd(gwContact);

        GetItemsResponse res=getContactNotes();

        try
        {
            mRes = m_gwMain.getGroupWiseService().modifyItemRequest(
                res.getItem().getItem()[0].getId(), null, iChanges,0,
                m_gwMain.getGroupWiseServiceId(), false);
        }
        catch (RemoteException ex)
        {
            ex.printStackTrace();
        }
    }
}

```

## Modification of a ContactNote

Below is a trace modifying an existing ContactNote on a contact. Notice the contact id is just below the modifyitemrequest method call. The ContactNote id is in the updates > update > contactnotes > contactnote section.

```

<ns0:modifyItemRequest>
<ns0id>48FEE952.domain1.pol.104.16A6163.1.4A.1@56:4847C63F.domain1.pol.104.16A6163.1.3.1@53</ns0:id>
<ns0:updates>
<ans1:update xmlns:ans1="http://schemas.novell.com/2005/01/GroupWise/types" xsi:type="ans1:Contact">
<ans1:version>0</ans1:version>
<ans1:contactNotes>
<ans1:contactNote>
<ans1:id>4906DFDF.domain1.pol.100.16A6163.1.1413.1@2:A.domain1.pol.100.0.1.0.1@19</ans1:id>
<ans1:attachment>
<ans1:id
itemReference="false">4906DFDF.domain1.pol.100.16A6163.1.1412.1@65</
ans1:id>
<ans1:contentType>text/plain</ans1:contentType>
<ans1:size>0</ans1:size>
<ans1:data>TW9kaWZpY2F0aW9ulHZpYSBTT0FQIC0gMQ==</ans1:data>
</ans1:attachment>
</ans1:contactNote>
</ans1:contactNotes>
</ans1:update>
</ns0:updates>
</ns0:modifyItemRequest>

```

## Deletion of a ContactNote

Below is a trace deleting an existing ContactNote on a contact. Notice the contact id is just below the modifyitemrequest method call. The ContactNote id is in the updates > delete > contactnotes > contactnote section.

```
<ns0:modifyItemRequest>
<ns0:id>48FEE952.domain1.pol.104.16A6163.1.4A.1@56:4847C63F.domain1.pol.10
4.16A6163.1.3.1@53</ns0:id>
<ns0:updates>
<ans1:delete xmlns:ans1="http://schemas.novell.com/2005/01/GroupWise/
types" xsi:type="ans1:Contact">
<ans1:version>0</ans1:version>
<ans1:contactNotes>
<ans1:contactNote>
<ans1:id>4906DFDF.domain1.pol.100.16A6163.1.1413.1@2:A.domain1.pol.100.0.1
.0.1@19</ans1:id>
</ans1:contactNote>
</ans1:contactNotes>
</ans1:delete>
</ns0:updates>
<ns0:recurrenceAllInstances>0</ns0:recurrenceAllInstances>
</ns0:modifyItemRequest>
```

## Display Settings

You can get the Display Settings for the Windows client. There are several different types of display settings that can be retrieved.

---

**NOTE:** In GroupWise 8.0 SP2, the noDisplay keyword was added to the view so the folder display settings are not returned when getting the folder(s) object. The display settings are not needed in normal use. There is thus no reason to get the settings for most cases. This reduces some of the SOAP traffic.

---

### Folder

Returns the folder display settings list. In the Windows client, you can view this same information by selecting View > Display Settings > Select from the main toolbar.

To get the Folder display settings, you need to call getItemRequest with "Folder" as the container. For example, getItemRequest ("Folder",view, null, null, 0, session, false);.

### FolderDefault

Returns the folder list default display settings. In other words, it shows all the default settings for the system folders.

To get the FolderDefault display settings, you need to call getItemRequest with "FolderDefaults" as the container. For example, getItemRequest ("FolderDefault", view, null, null, 0, session, false);.

## Panel

Returns the panels defined on the home folder for the user.

To get the Panel display settings, you need to call `getItemsRequest` with “Panel” as the container. For example, `getItemsRequest (“Panel”, view, null, null, 0, session, false);`.

## PanelTemplates

Returns all the available panels and their settings. In the Windows client, you can view this same information by selecting View > Display Settings > Panels from the main toolbar.

To get the panelTemplates display settings, you need to call `getItemsRequest` with “PanelTemplates” as the container. For example, `getItemRequest (“PanelTemplates”, view, null, null, 0, session, false);`.

# Item Modification

Received and sent items are for the most part read-only. There are a few fields that you can make some personalization to the original item. You can change the following fields:

- ♦ Subject
- ♦ Categories
- ♦ xField
- ♦ Personal attachment
- ♦ Contact
- ♦ Alarm (Appointment items)
- ♦ acceptLevel (Appointment items)
- ♦ ChecklistInfo
- ♦ Custom fields

Sent appointments and received appointment (where you are the sender) have other fields that can be changed. If you modify the start time, end time or place of one of these appointments, the POA does extra logic. It retracts the original appointment and send out a new appointment with the updated times or place.

It is also possible to change the recipients. This applies if you are just adding or removing recipients (not changing the time or place of the appointment). In this case, you supply a new Distribution element in the `<add>` element of the `ItemChanges` for the `modifyItemRequest`. You add the recipients you want in the final appointment. If there are recipients in the original appointment that are not in the updated appointment, the appointment is retracted from those users. If there are new recipients in the updated appointment, the appointment is sent to those users. If there are recipients in the old appointment and new appointment, the appointment is not sent to them. You also have to supply the updated to, cc and / or bc elements in the Distribution element.

The modifying of sent and received appointments was added in GroupWise 8.0 SP2.

In GroupWise 2012, you can also use the `resendRequest`. This is the same concept as in the GroupWise client. You can make extensive changes and then send the new item using `sendItemRequest`. You need to manually retract the original item using the `retractRequest`.

## Example

```
<id>4D3E9ACB.domain.P01.100.1776172.1.25C9C.1@4:A.domain.P01.100.0.1.0.1@1
9
  </id>
  <updates>
    <add type="Appointment">
      <distribution>
        <to>Preston Stepehnson; Jay Parker</to>
        <recipients>
          <recipient type="Recipient">
            <displayName>Preston Stephenson</displayName>
            <email>pstehenson@prestons.novell.com</email>
            <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
          </recipient>
          <recipient type="Recipient">
            <displayName>Jay Parker</displayName>
            <email>jparker@prestons.novell.com</email>
            <uuid>C974CA00-0B55-0000-B976-8E90E5B002BD</uuid>
          </recipient>
        </recipients>
      </distribution>
    </add>
  </updates>
</modifyItemRequest>

<modifyItemResponse>
  <modified>2011-01-25T17:51:11Z</modified>
  <status>
    <code>0</code>
  </status>
</modifyItemResponse>
```





# 2 Methods

You must call [loginRequest](#) before calling any other methods. The loginRequest method returns a session key in the loginResponse that is required by all other methods during that session.

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

- ♦ [“acceptRequest”](#) on page 76
- ♦ [“acceptShareRequest”](#) on page 78
- ♦ [“addItemRequest”](#) on page 80
- ♦ [“addItemsRequest”](#) on page 81
- ♦ [“addMembersRequest”](#) on page 82
- ♦ [“archiveRequest”](#) on page 83
- ♦ [“closeFreeBusySessionRequest”](#) on page 86
- ♦ [“completeRequest”](#) on page 87
- ♦ [“createCursorRequest”](#) on page 88
- ♦ [“createItemRequest”](#) on page 90
- ♦ [“createItemsRequest”](#) on page 92
- ♦ [“createJunkEntryRequest”](#) on page 93
- ♦ [“createNotifyRequest”](#) on page 94
- ♦ [“createProxyAccessRequest”](#) on page 96
- ♦ [“createSignatureRequest”](#) on page 98
- ♦ [“declineRequest”](#) on page 99
- ♦ [“delegateRequest”](#) on page 100
- ♦ [“destroyCursorRequest”](#) on page 102
- ♦ [“executeRuleRequest”](#) on page 103
- ♦ [“forwardRequest”](#) on page 104
- ♦ [“getAccountListRequest”](#) on page 106
- ♦ [“getAddressBookListRequest”](#) on page 107
- ♦ [“getArchiveItemsRequest”](#) on page 109
- ♦ [“getAttachmentRequest”](#) on page 111
- ♦ [“getCategoryListRequest”](#) on page 113
- ♦ [“getContactHistoryFilterRequest”](#) on page 115
- ♦ [“getCustomListRequest”](#) on page 127
- ♦ [“getDeltasRequest”](#) on page 128
- ♦ [“getDeltaInfoRequest”](#) on page 131

- ♦ “getDiskSpaceUsageRequest” on page 133
- ♦ “getDocumentTypeListRequest” on page 134
- ♦ “getFolderRequest” on page 136
- ♦ “getFolderListRequest” on page 138
- ♦ “getFreeBusyRequest” on page 143
- ♦ “getItemRequest” on page 146
- ♦ “getItemsRequest” on page 148
- ♦ “getJunkEntriesRequest” on page 154
- ♦ “getJunkMailSettingsRequest” on page 156
- ♦ “getLibraryItemRequest” on page 158
- ♦ “getLibraryListRequest” on page 160
- ♦ “getMemberOfRequest” on page 161
- ♦ “getNotifyListRequest” on page 163
- ♦ “getProxyAccessListRequest” on page 165
- ♦ “getProxyListRequest” on page 167
- ♦ “getQuickMessagesRequest” on page 169
- ♦ “getRuleListRequest” on page 173
- ♦ “getSettingsRequest” on page 176
- ♦ “getSignaturesRequest” on page 184
- ♦ “getStringRequest” on page 186
- ♦ “getTimestampRequest” on page 187
- ♦ “getTimezoneListRequest” on page 189
- ♦ “getUnreadRequest” on page 191
- ♦ “getUserListRequest” on page 195
- ♦ “loginRequest” on page 197
- ♦ “logoutRequest” on page 202
- ♦ “markPrivateRequest” on page 203
- ♦ “markReadRequest” on page 204
- ♦ “markUnPrivateRequest” on page 205
- ♦ “markUnReadRequest” on page 206
- ♦ “modifyItemRequest” on page 207
- ♦ “modifyItemsRequest” on page 211
- ♦ “modifyJunkEntryRequest” on page 212
- ♦ “modifyJunkMailSettingsRequest” on page 213
- ♦ “modifyPasswordRequest” on page 214
- ♦ “modifyNotifyRequest” on page 215
- ♦ “modifyProxyAccessRequest” on page 216
- ♦ “modifySettingsRequest” on page 218

- ♦ “modifySignaturesRequest” on page 219
- ♦ “moveItemRequest” on page 220
- ♦ “moveItemsRequest” on page 222
- ♦ “positionCursorRequest” on page 224
- ♦ “purgeDeletedItemsRequest” on page 226
- ♦ “purgeRequest” on page 227
- ♦ “readCursorRequest” on page 228
- ♦ “removeCustomDefinitionRequest” on page 230
- ♦ “removeItemRequest” on page 232
- ♦ “removeItemsRequest” on page 233
- ♦ “removeJunkEntryRequest” on page 235
- ♦ “removeMembersRequest” on page 236
- ♦ “removeProxyAccessRequest” on page 237
- ♦ “removeNotifyRequest” on page 238
- ♦ “removeSignatureRequest” on page 239
- ♦ “replyRequest” on page 240
- ♦ “resendRequest” on page 242
- ♦ “resolveRequest” on page 248
- ♦ “restoreItemRequest” on page 249
- ♦ “retractRequest” on page 250
- ♦ “sendItemRequest” on page 252
- ♦ “setTimestampRequest” on page 254
- ♦ “startFreeBusySessionRequest” on page 256
- ♦ “streamedSearchRequest” on page 258
- ♦ “stubItemRequest” on page 262
- ♦ “unacceptRequest” on page 263
- ♦ “unarchiveRequest” on page 264
- ♦ “uncompleteRequest” on page 265
- ♦ “updateversionStatusRequest” on page 266

# acceptRequest

Accepts items.

## Request

```
<acceptRequest>
  <items/>
  <comment/>
  <acceptLevel/>
  <recurrenceAllInstances/>
  <container/>
  <alarm/>
</acceptRequest>
```

## Response

```
<acceptResponse>
  <status>
    <code>0</code>
  </status>
</acceptResponse>
```

## Elements

### items

Specifies a list of item IDs to be accepted.

### comment

Specifies text that the sender of the item can view in the send item properties.

### acceptLevel

Specifies how you want your appointments to appear in busy searches:

- ♦ Free
- ♦ Tentative
- ♦ Busy
- ♦ Out of Office

### recurrenceAllInstances

Specifies that you want the accept and its options to be applied to all recurring items if you pass the recurrenceKey.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

**container**

Specifies the calendar/container as the destination for the accepted calendar item.

**alarm**

Specifies the alarm value.

**Example**

```
<acceptRequest>
  <items>
    <item>40ADFA62.AutoDomain.AutoP01.100.16F3471.1.FEC.1@4:7.
      AutoDomain.AutoP01.100.0.1.0.1@16</item>
  </items>
  <comment>This is a comment</comment>
  <acceptLevel>Tentative</acceptLevel>
  <recurrenceAllInstances>532</recurrenceAllInstances>
</acceptRequest>
```

# acceptShareRequest

Accepts a shared folder notification or a shared personal address book notification.

## Request

```
<acceptShareRequest>
  <id/>
  <name/>
  <container/>
  <description/>
  <sequence/>
</acceptShareRequest>
```

## Response

```
<acceptShareResponse>
  <id/>
  <status/>
</acceptShareResponse>
```

## Elements

### id

Specifies the ID of the shared folder notification to accept.

### name

Specifies the name of the folder to create. If no name is given, the original name from the notification message is used.

### container

Specifies the container ID of the parent folder where the new folder is created. If the container is not specified, the cabinet folder is the default container.

### description

(optional) Specifies a detailed description of the shared folder. This is text supplied by the creator of the shared folder.

### sequence

Specifies where to place the new folder in the sequence of folders.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### id

Returns the ID for the newly created folder.

### status

Returns the success or failure of the method.

## Example

```
<acceptShareRequest>
  <id>44F71EE4.domain1.pol.100.16E3837.1.F86.1@37:7.
    domain1.pol.100.0.1.0.1@16</id>
  <name>Shared Folder for project A</name>
  <container>C.domain1.pol.100.0.1.0.1@21</container>
  <description>Shared Folder Description</description>
</acceptShareRequest>

<acceptShareResponse>
  <id>44F71F5C.domain1.pol.100.16E3837.1.F87.1@34:44F71EE1.
    domain1.pol.100.1676834.1.766.1</id>
  <status>
    <code>0</code>
  </status>
</acceptShareResponse>
```

# addItemRequest

Adds an existing address book item (such as a contact, resource, organization, or group) to a different personal address book. The item is also called a linked item because the original address book item is not deleted. The user's database has one item that is referenced by many personal address books.

## Request

```
<addItemRequest>
  <container/>
  <id/>
</addItemRequest>
```

## Response

```
<addItemResponse>
  <status/>
</addItemResponse>
```

## Elements

### container

Specifies the ID of the destination personal address book.

### id

Specifies the item ID that is to be linked to another personal address book.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<addItemRequest>
  <container>44F83F1D.domain1.pol.104.16E3837.1.F8.1@53<container>
  <id>44F83F50.domain1.pol.104.16E3837.1.FA.1@56:42C510EA.
    domain1.pol.104.16E3837.1.3.1@53<id>
</addItemRequest>
```



# addItemsRequest

Adds existing address book items (such as contacts, resources, organizations, and groups) to a different personal address book. The item is also called a linked item because the original address book items are not deleted. The user's database has several items that are referenced by many personal address books.

## Request

```
<addItemsRequest>
  <container/>
  <items/>
</addItemsRequest>
```

## Response

```
<addItemsResponse>
  <status/>
</addItemsResponse>
```

## Elements

### container

Specifies the ID of the destination personal address book.

### items

Specifies the items that are to be linked to another personal address book.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<addItemsRequest>
  <container>44F83F1D.domain1.pol.104.16E3837.1.F8.1@53</container>
  <items>
    <item>44F855F7.domain1.pol.104.16E3837.1.FD.1@57:42C510EA.
      domain1.pol.104.16E3837.1.3.1@53</item>
    <item>44F855E9.domain1.pol.104.16E3837.1.FC.1@55:42C510EA.
      domain1.pol.104.16E3837.1.3.1@53</item>
  </items>
</addItemsRequest>
```

# addMembersRequest

Adds existing address book items (such as contacts, resources, organizations, and groups) to an existing personal address book group.

## Request

```
<addMembersRequest>
  <container/>
  <members/>
</addMembersRequest>
```

## Response

```
<addMembersResponse>
  <status/>
</addMembersResponse>
```

## Elements

### container

Specifies the ID of the group.

### members

Points to existing personal address book items to add to the group that is specified in the container element.

The GroupMember element provides parameters that are not used. For example, the name element can be set in the GroupMember element. However, the item name on the actual personal address book item is not changed.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<addMembersRequest>
  <container>44EB3142.domain1.pol.104.16E3837.1.F0.1@58:44F83F1D.
    domain1.pol.104.16E3837.1.F8.1@53</container>
  <members>
    <member>
      <id>44EB3142.domain1.pol.104.16E3837.1.F0.1@58:42C510EA.
        domain1.pol.104.16E3837.1.3.1@53</id>
      <distType>T0</distType>
    </member>
  </members>
</addMembersRequest>
```

# archiveRequest

Archives item(s) to an archive database. The archive path needs to be a path that the POA has access to (or can reach).

## Request

```
<archiveRequest>
  <items/>
  <path/>
</archiveRequest>
```

## Response

```
<archiveResponse>
  <path/>
  <status/>
</archiveResponse>
```

## Elements

### items

Specifies the items to archive.

### path

On input, specifies the path to use to create the archive directory. If a path is not specified, the POA oftemp directory is used.

On output, returns the path to the archive directory is used.

### code

Returns the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<archiveRequest>
  <items>
    <item>4D37DDE2.domain.PO3.100.16A7363.1.23E.1@1:
      7.domain.PO3.100.0.1.0.1@16</item>
  </items>
  <path>/shr/archive</path>
</archiveRequest>

<archiveResponse>
  <path>/shr/archive/ofrawarc</path>
  <status>
    <code>0</code>
  </status>
</archiveResponse>

<getArchiveItemsRequest>
  <container>folders</container>
  <count>0</count>
  <path>/shr/archive/</path>
</getArchiveItemsRequest>

<getArchiveItemsResponse>
  <items>
    <item type="Mail">

<id>4D37DE41.domain.PO3.100.16A7363.1.17.1@1:7.domain.PO3.100.0.1.0.1@16</
id>
  <sid>23</sid>
  <version>3</version>
  <modified>2011-01-20T14:03:29Z</modified>
  <container>7.domain.PO3.100.0.1.0.1@16</container>
  <created>2011-01-20T14:01:54Z</created>
  <status>
    <opened>1</opened>
    <read>1</read>
  </status>
  <msgId>4D37DE41.domain.PO3.200.2000000.1.3.1</msgId>
  <clientMessageId>4D384052.CD9 : 4 : 31797</clientMessageId>
  <source>received</source>
  <delivered>2011-01-20T14:01:54Z</delivered>
  <security>Normal</security>
  <subject>to archive</subject>
  <distribution>
    <from>
      <displayName>Layne Izatt</displayName>
      <email>lizatt@prestons.provo.novell.com</email>
      <uuid>FBC97000-100A-0000-90B7-0659E3122263</uuid>
    </from>
    <to>Layne Izatt</to>
    <sendoptions>
      <statusTracking>None</statusTracking>
    </sendoptions>
  </distribution>
```

```

    <options>
      <priority>Standard</priority>
    </options>
    <size>936</size>
    <threading>
<id>4D37DDE2.domain.PO3.200.2000004.1.5D.1:
  4D37DE41.domain.PO3.200.2000000.1.3.1</id>
    <parent>4D37DDE2.domain.PO3.200.2000004.1.5D.1</parent>
    </threading>
    <retentionModified>2011-01-20T14:03:29Z</retentionModified>
  </item>
</items>
<status>
  <code>0</code>
</status>
</getArchiveItemsResponse>

<unarchiveRequest>
  <items>
    <item>4D37DE41.domain.PO3.100.16A7363.1.17.1@1:
      7.domain.PO3.100.0.1.0.1@16</item>
  </items>
  <path>/shr/archive</path>
</unarchiveRequest>

<unarchiveResponse>
  <path>/shr/archive/ofrawarc</path>
  <status>
    <code>0</code>
  </status>
</unarchiveResponse>

```

# closeFreeBusySessionRequest

Frees server-side resources that are created by calling [startFreeBusySessionRequest](#). `startFreeBusySessionRequest` returns a `freeBusySessionId` that points to the server-side free/busy resource. This method should be called when all the free/busy information has been returned.

## Request

```
<closeFreeBusySessionRequest>
  <freeBusySessionId/>
</closeFreeBusySessionRequest>
```

## Response

```
<closeFreeBusySessionResponse>
  <status/>
</closeFreeBusySessionResponse>
```

## Elements

### **freeBusySessionId**

Points to the server-side FreeBusy resource that was returned in the `startFreeBusySessionResponse`.

### **code**

Returns the error number related to the event. 0 indicates that the request was successful.

### **status**

Returns the success or failure of the method.

## Example

```
<closeFreeBusySessionRequest>
  <freeBusySessionId>354</freeBusySessionId>
</closeFreeBusySessionRequest>

<closeFreeBusySessionResponse>
  <status>
    <code>0</code>
  </status>
</closeFreeBusySessionResponse>
```

# completeRequest

Marks a task as complete.

## Request

```
<acceptRequest>
  <items/>
  <comment/>
  <acceptLevel/>
  <recurrenceAllInstances/>
</acceptRequest>
```

## Response

```
<acceptResponse>
  <status>
    <code/>
  </status>
</acceptResponse>
```

## Elements

### items

Specifies a list of item IDs to accept.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<completeRequest>
  <items>
    <item>44FD563E.domain1.pol.100.16E3837.1.F8B.1@3:
      7.domain1.pol.100.0.1.0.1@16
    </item>
  </items>
</completeRequest>
```

# createCursorRequest

Creates a server-side cursor resource for retrieving large amounts of container data in small chunks and allows you to read containers (such as a mailbox folder or an address book). If a createCursorRequest is successful, destroy the server-side cursor resource by calling destroyCursorRequest. For more information please see [Getting Items](#).

## Request

```
<createCursorRequest>
  <container/>
  <view/>
  <filter/>
</createCursorRequest>
```

## Response

```
<createCursorResponse>
  <cursor/>
  <status/>
</createCursorResponse>
```

## Elements

### container

Specifies the container's ID. If the container is empty, the cursor reads all folders in the user's database (with the exception of the shared folders shared with the user, the query folders and the Trash folder).

### view

Specifies the elements that are returned for each item. The view reduces the amount of data returned. If a view is not specified, all elements are returned.

### filter

Specifies restrictions on the items to read.

### cursor

Specifies the server-side cursor resource. The cursor ID can be used with other cursor methods such as [positionCursorRequest](#), [readCursorRequest](#), and [destroyCursorRequest](#).

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.



## Example

```
<createCursorRequest>
  <container>19.domain1.pol.100.0.1.0.1@30</container>
  <view>id messageId subject message recipients</view>
  <filter>
    <element type="FilterEntry">
      <op>gt</op>
      <field>created</field>
      <value>2012-08-15T00:00:00Z</value>
    </element>
  </filter>
</createCursorRequest>

<createCursorResponse>
  <cursor>1195551863</cursor>
  <status>
    <code>0</code>
  </status>
</createCursorResponse>
```

# createItemRequest

Creates mail, appointment, tasks, notes, and phone items in a mailbox without sending out the item to other users. If you need to send personal, posted, or distributed items, call [sendItemRequest](#).

## Request

```
<createItemRequest>
  <item/>
  <notification/>
</createItemRequest>
```

## Response

```
<createItemResponse>
  <id/>
  <status/>
</createItemResponse>
```

## Elements

### item

Specifies the item to create.

### notification

Specifies the information that is used in a shared folder or personal address book notification message.

### id

On a successful response, specifies the ID of the newly created item. Recurring items return an ID for each item that was created.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

The createItemRequest method is used to create the following item types:

- ♦ Address Book Items:
  - ♦ Contacts
  - ♦ Groups
  - ♦ Organizations
  - ♦ Personal Address Books
  - ♦ Resources

- ♦ Categories
- ♦ Folders
- ♦ Rules
- ♦ Shared Folders
- ♦ Mail Items
  - ♦ Mail Messages
  - ♦ Appointments
  - ♦ Tasks
  - ♦ Notes
  - ♦ Phone Messages

## Example

```

<createItemRequest>
  <item types="Contact">
    <container>406901A0.provo.research.104.16C3230.1.3.1@53</container>
    <categories types:primary="1.provo.research.100.0.1.0.1@61"
      <category>4.provo.research.100.0.1.0.1@61</category>
      <category>1.provo.research.100.0.1.0.1@61</category>
    </categories>
    <fullName>
      <displayName>John Doe</displayName>

      <firstName>John</firstName>
      <lastName>Doe</lastName>
    </fullName>
    <emailList primary="jdoe@novell.com"
      <email>jdoe@novell.com</email>
    </emailList>
    <email>jdoe@novell.com</email>
  </types:item>
</createItemRequest>

<createItemResponse>
  <id>407D48DF.provo.research.104.16C3230.1.7D.1@56</id>
  <status>
    <code>0</code>
  </status>
</createItemResponse>

```

## **createItemsRequest**

Is similar to [createItemRequest](#). The only difference is that createItemsRequest can create more than one item at a time.

# createJunkEntryRequest

Provides access to the junk mail handling, block, and trust lists for a user.

## Request

```
<createJunkEntryRequest>
  <entry/>
</createJunkEntryRequest>
```

## Response

```
<createJunkEntryResponse>
  <id/>
  <status>
    <code/>
  </status>
</createJunkEntryResponse>
```

## Elements

### entry

Specifies the new junk mail handling entry.

### id

On a successful response, specifies the ID of the newly created junk mail entry.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<createJunkEntryRequest>
  <entry>
    <match>user2@block.net</match>
    <matchType>email</matchType>
    <listType>block</listType>
    <lastUsed />
    <version>0</version>
  </entry>
</createJunkEntryRequest>

<createJunkEntryResponse>
  <id>44FE8B4B.domain1.pol.104.16E3837.1.106.1@66</id>
  <status>
    <code>0</code>
  </status>
</createJunkEntryResponse>
```

# createNotifyRequest

Creates a notify entry. In the GroupWise Windows client, you can set up to be notified for another persons alarms and deliveries (see “Being Notified of Someone Else's Messages” ([http://www.novell.com/documentation/groupwise2012/gw2012\\_guide\\_userwin/data/aai3c1s.html](http://www.novell.com/documentation/groupwise2012/gw2012_guide_userwin/data/aai3c1s.html))). The createNotifyRequest, getNotifyListRequest, modifyNotifyRequest and removeNotifyRequest are used to manage these settings. --For GroupWise 8.0 SP1 and later.

## Request

```
<createNotifyRequest>
  <entry/>
</createNotifyRequest>
```

## Response

```
<createNotifyResponse>
  <id/>
  <status/>
</createNotifyResponse>
```

## Elements

### entry

Specifies the user and notification options.

### id

On a successful response, specifies the ID of the NotifyEntry.

### code

Returns the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the request.

## Example

```
<createNotifyRequest>
  <entry type="NotifyEntry">
    <displayName>Jay Parker</displayName>
    <email>jparker@prestons.provo.novell.com</email>
    <uuid>C974CA00-0B55-0000-B976-8E90E5B002BD</uuid>
    <rights>
      <alarm>true</alarm>
      <notification>true</notification>
    </rights>
  </entry>
</createNotifyRequest>

<createNotifyResponse>
  <id>4D3D6590.domain.P01.100.1776172.1.25C8C.1@78</id>
  <status>
    <code>0</code>
  </status>
</createNotifyResponse>
```

# createProxyAccessRequest

Allows a user to grant access to his or her mailbox to another user.

## Request

```
<createProxyAccessRequest>  
  <entry/>  
</createProxyAccessRequest>
```

## Response

```
<createProxyAccessResponse>  
  <id/>  
  <status/>  
</createProxyAccessResponse>
```

## Elements

### entry

Specifies the user to grant access to and the access rights he or she has to the mailbox.

### id

On a successful response, specifies the ID of the AccessRightEntry.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

The user can limit the type of access a proxy has by using access rights. For example, users could specify that another user has read or write access to mail, appointments, tasks, notes, alarms, notifications, options, and private items.



## Example

```
<createProxyAccessRequest>
  <entry>
    <displayName>u4</displayName>
    <email>u4@phantom.com</email>
    <uuid>66301550-175B-0000-890F-6E00D5004E00</uuid>
    <appointment>
      <read>1</read>
      <write>1</write>
    </appointment>
    <mail>
      <read>1</read>
      <write>1</write>
    </mail>
    <misc>
      <alarms>1</alarms>
      <notify>1</notify>
      <readHidden>1</readHidden>
      <setup>1</setup>
    </misc>
    <note>
      <read>1</read>
      <write>1</write>
    </note>
    <task>
      <read>1</read>
      <write>1</write>
    </task>
  </entry>
</createProxyAccessRequest>
```

# createSignatureRequest

Creates a new signature for a user. It returns a signature ID after it successfully completes. HTML signatures are stored as a MIME type, according to the MIME (RFC822) standard for storing signatures.

## Request

```
<createSignatureRequest>
  <signature/>
</createSignatureRequest>
```

## Response

```
<createSignatureResponse>
  <status/>
</createSignatureResponse>
```

## Elements

### signature

Specifies the signature to be created.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

# declineRequest

Declines items.

## Request

```
<declineRequest>
  <items/>
  <comment/>
  <recurrenceAllInstances/>
</declineRequest>
```

## Response

```
<declineResponse>
  <status>
    <code/>
  </status>
</declineResponse>
```

## Elements

### items

Specifies a list of item IDs to be declined.

### comment

Specifies text that the sender of the item can view in the properties of the sent item.

### recurrenceAllInstances

Specifies the recurrenceKey so that the decline and its options can be applied to all the recurring items.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<declineRequest>
  <items>
    <item>44FED4BA.domain1.pol.100.16E3837.1.F9D.1@4:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
  <comment>asdf</comment>
  <recurrenceAllInstances>36167</recurrenceAllInstances>
</declineRequest>
```

# delegateRequest

Delegates calendar items (appointments, notes, and tasks) to other users.

## Request

```
<delegateRequest>
  <id type="types:uid"/>
  <commentToOrganizer type="string"/>
  <commentToDelegatee type="string"/>
  <distribution type="types:Distribution"/>
  <recurrenceAllInstances type="recurrenceKey"/>
</delegateRequest>
```

## Response

```
<delegateResponse>
  <status type="types:Status"/>
</delegateResponse>
```

## Elements

### id

Specifies the ID of the item to be delegated.

### commentToOrganizer

Specifies a comment that the sender of the original calendar item can view in the properties of the sent item.

### commentToDelegatee

Specifies a comment to the delegatee of the item. This element is not used by any GroupWise client.

### distribution

Specifies the recipients for the delegated calendar item.

### recurrenceAllInstances

Specifies the recurrenceKey so that the delegate and its options can be applied to all the recurring items.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<delegateRequest>
  <id>44FED73B.domain1.pol.100.16E3837.1.FAA.1@2:7.
    domain1.pol.100.0.1.0.1@16</id>
  <commentToOrganizer>Comment to Organizer</commentToOrganizer>
  <commentToDelegatee>Comment to Delegatee</commentToDelegatee>
  <distribution>
    <recipients>
      <recipient>
        <displayName>u3</displayName>
        <email>u3@phantom.com</email>
        <uuid>54D55590-175B-0000-890F-6E00D5004E00</uuid>
      </recipient>
    </recipients>
  </distribution>
</delegateRequest>
```

# destroyCursorRequest

Deletes the server-side cursor resource that was created by the [createCursorRequest](#) method. For more information please see [Getting Items](#).

## Request

```
<destroyCursorRequest>
  <container type="types:uid"/>
  <cursor type="int"/>
</destroyCursorRequest>
```

## Response

```
<destroyCursorResponse>
  <status type="types:Status"/>
</destoryCursorResponse>
```

## Elements

### container

Specifies the container for the server-side cursor.

### cursor

Specifies the identifier for the server-side cursor resource that is to be deleted.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<destroyCursorRequest>
  <container>19.domain1.pol.100.0.1.0.1@30</container>
  <cursor>1195457801</cursor>
</destroyCursorRequest>
```

# executeRuleRequest

Immediately runs or executes a previously defined rule.

## Request

```
<executeRuleRequest>
  <id type="types:uid"/>
</executeRuleRequest>
```

## Response

```
<executeRuleResponse>
  <status type="types:Status"/>
</executeRuleResponse>
```

## Elements

### id

Points to the rule to be executed.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<executeRuleRequest>
  <id>44630E88.domain1.po1.100.16E3837.1.B5E.1@10</id>
</executeRuleRequest>
```

# forwardRequest

Forwards an existing item to other users.

## Request

```
<forwardRequest>
  <id type="types:uid"/>
  <view type="string"/>
  <embed type="boolean"/>
</forwardRequest>
```

## Response

```
<forwardResponse>
  <item type="types:Item"/>
  <status type="types:Status"/>
</forwardResponse>
```

## Elements

### id

Specifies the ID of the item you want to forward.

### view

Specifies if you want the original message body returned in the forward method. If you do, supply “message” or “message/RTF” in the view.

### embed

Specifies if the original item is to be sent as an attachment in the new mail item. If you do, pass True in the embed element in the forwardRequest. Otherwise, pass False.

If you set embed to true, you need to make sure the view on forwardRequest does not contain the "attachments" keyword. The forwardResponse will then send down an attachment structure that needs to be added to the new item.

### item

Specifies a returned item that can be modified and passed to the sendRequest method to complete the forwardRequest.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.



## Remarks

Forwarding a message is a two-step process.

The first step is to call `forwardRequest` with the ID of the item that you want to forward. A new mail item is then returned with the original subject and other fields defined in the view. You can modify this returned item. For example, the caller might want to prepend "Fwd:" to the beginning of the subject. The message body and other fields can also be modified and attachments added. The `linkInfo` element is returned with this item.

The second step is to call `sendRequest` with the modified mail item. The `linkInfo` element needs to be passed unchanged in the `sendRequest`. The `linkInfo` element is necessary for the POA to link the original item with the new item using the `sendRequest`.

## Example

```
<forwardRequest>
<id>44FEDD14.domain1.po1.100.16E3837.1.FB2.1@4:7.domain1.po1.100.0.1.0.1@1
6</id>
<view>subject id message</view>
<embed>false</embed>
</forwardRequest>

<forwardResponse>
<item>
<source>sent</source>
<subject>New Appt 1</subject>
<message>
<part contentType="text/plain"
length="64">YXNkZgogCiAKVXNlciAxCiAgTm92ZWxsIEluYy4gIAogICA4NjEtMTEzMCAgIA
o=</part>
</message>
<attachments>
<attachment>
<id itemReference="1">4018</id>
<name>FILE</name>
<contentType>Appointment</contentType>
<hidden>1</hidden>
</attachment>
</attachments>
<link>
<id>0:4018</id>
<type>forward</type>
</link>
</item>
<status>
<code>0</code>
</status>
</forwardResponse>
```

# getAccountListRequest

Returns the list of accounts.

## Request

```
<"getAccountListRequest">
```

## Response

```
<getAccountListResponse>  
  <account type="types:ItemList"/>  
  <status type="types:Status"/>  
</getAccountListResponse>
```

## Elements

### accounts

A list of accounts.

### status

Returns the success or failure of the method.

# getAddressBookListRequest

Returns the list of address books.

## Request

```
<getAddressBookListRequest>
  <name type="string"/>
  <view type="string"/>
</getAddressBookListRequest>
```

## Response

```
<getAddressBookListResponse>
  <books type="types:AddressBookList"/>
  <status type="types:Status"/>
</getAddressBookListResponse>
```

## Elements

### books

Returns the list of personal address books and the ID of the GroupWise Address Book.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

### name

Specify the name of the book to return.

### view

Specifies the elements that are returned for each item. The view reduces the amount of data returned. If a view is not specified, all elements are returned.

## Example

```
<getAddressBookListRequest />

<getAddressBookListResponse>
  <books>
    <book>
      <id>42C510EA.domain1.pol.104.16E3837.1.1.1@53</id>
      <name>Frequent Contacts</name>
      <version>2</version>
      <modified>2005-07-01T15:46:18Z</modified>
      <isPersonal>1</isPersonal>
      <isFrequentContacts>1</isFrequentContacts>
    </book>
    <book>
      <id>GroupWiseSystemAddressBook@52</id>
      <name>Novell GroupWise Address Book</name>
    </book>
    <book>
      <id>44F83F1D.domain1.pol.104.16E3837.1.F8.1@53</id>
      <name>New Book</name>
      <version>2</version>
      <modified>2012-09-01T20:09:33Z</modified>
      <isPersonal>1</isPersonal>
    </book>
    <book>
      <id>42C510EA.domain1.pol.104.16E3837.1.3.1@53</id>
      <name>ul</name>
      <version>9</version>
      <modified>2012-08-23T15:53:54Z</modified>
      <isPersonal>1</isPersonal>
    </book>
  </books>
</getAddressBookListResponse>
```

# getArchiveItemsRequest

Returns items from the specified archive database.

## Request

```
<container type="types:uid" />
<view type="types:View" />
<filter type="types:Filter" />
<items type="types:ItemRefList" />
<count type="int" />
<path type="string" />
</getArchiveItemsRequest>
```

## Response

```
<getArchiveItemsResponse>
  <items type="types:Items" />
  <status type="types:Status" />
</getArchiveItemsResponse>
```

## Elements

### container

(optional) Specifies the folder, address book, or library container ID. If the container is not specified, the filter is used. If the filter is not present, an error is returned.

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned.

### filter

Specifies the items to return based on a filter.

### items

On input, specifies a list of item ids to return.

On return, returns the specified items.

### count

Specifies the number of items to return. If the count element is -1, all items are returned.

### path

Specifies the path to the archive directory. If not specified, the POA oftemp directory is used.

### code

Specifies the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

See [archiveRequest](#).

# getAttachmentRequest

Returns the selected file attachment. It does not read embedded GroupWise items such as mail messages or appointments.

## Request

```
<getAttachmentRequest>
  <id type="types:uid"/>
  <offset type="int"/>
  <length type="int"/>
  <flags type="types:AttachmentFlags"/>
</getAttachmentRequest>
```

## Response

```
<getAttachmentResponse>
  <part type="types:MessagePart"/>
  <status type="types:Status"/>
</getAttachmentResponse>
```

## Elements

### id

Specifies the ID of the attachment to read.

### offset

Specifies the starting position for reading the attachment.

### length

Specifies the number of bytes to read from the offset.

### part

Specifies the attachment.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

### flags

Specifies the attachment flags.

## Remarks

To retrieve the entire file in one read, specify an offset of 0 with a length of -1. If you have a large file, you can read the attachment in chunks by using the offset element to position the attachment reader and specifying the length of the chunk's size.

## Example

```
<getAttachmentRequest>
  <id>45003893.domain1.pol.200.20000CB.1.1530.1@45:45003893.
    domain1.pol.100.16E3837.1.FBA.1@1:7.
    domain1.pol.100.0.1.0.1@16</id>
  <offset>0</offset>
  <length>-1</length>
</getAttachmentRequest>

<getAttachmentResponse>
  <part length="164"offset="122">
    MS4gIFNhdmUgT09PIGFzIHRleHQgZmlsZXMgbmV4dCB0aW1lLg0KMi4gIEl
    uY2xlZGUgdGhlIEJlZXAgY2xpZW50DQozLiAgRG1zY3VzcyB2ZXJzaW9uaW5
    nLg0KNC4gIE5ldEJlYW5zIDUuMCBCZXRhIDINCjUuICA=</part>
  <status>
    <code>0</code>
  </status>
</getAttachmentResponse>
```



# getCategoryListRequest

Returns the list of categories for the user.

## Request

```
<getCategoryListRequest/>
```

## Response

```
<getCategoryListResponse>
  <categories type="types:CategoryList"/>
  <status type="types:Status"/>
</getCategoryListResponse>
```

## Elements

### Categories

Returns the list of categories defined by the user.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getCategoryListRequest/>

<getCategoryListResponse>
  <categories>
    <category>
      <id>4.AutoDomain.AutoP01.100.0.1.0.1@61</id>
      <name>Low priority</name>
      <version>1</version>
      <modified>2002-10-02T23:17:10Z</modified>
      <type>LowPriority</type>
      <color>12632256</color>
    </category>
    <category>
      <id>3.AutoDomain.AutoP01.100.0.1.0.1@61</id>
      <name>Urgent</name>
      <version>1</version>
      <modified>2002-10-02T23:17:10Z</modified>
      <type>Urgent</type>
      <color>255</color>
    </category>
    <category>
      <id>2.AutoDomain.AutoP01.100.0.1.0.1@61</id>
      <name>Follow-up</name>
```

```

        <version>1</version>
        <modified>2002-10-02T23:17:10Z</modified>
        <type>Follow-up</type>
        <color>33023</color>
    </category>
    <category>
        <id>1.AutoDomain.AutoPO1.100.0.1.0.1@61</id>
        <name>Personal</name>
        <version>2</version>
        <modified>2002-10-30T14:57:09Z</modified>
        <type>Personal</type>
        <color>16711680</color>
    </category>
    <category>
        <id>3D047D59.AutoDomain.AutoPO1.100.1363230.1.435.1@12</id>
        <name>Business</name>
        <version>6</version>
        <modified>2004-02-19T15:39:08Z</modified>
        <type>Normal</type>
    </category>
    <category>
        <id>3DC90830.AutoDomain.AutoPO1.100.1363230.1.511.1@12</id>
        <name>Home</name>
        <version>1</version>
        <modified>2002-11-06T19:16:48Z</modified>
        <type>Normal</type>
    </category>
    <category>
        <id>3DBF94E2.AutoDomain.AutoPO1.100.1363230.1.4F9.1@12</id>
        <name>Meeting</name>
        <version>3</version>
        <modified>2004-02-19T15:39:08Z</modified>
        <type>Normal</type>
        <color>16711680</color>
    </category>
</categories>
<status>
    <code>0</code>
</status>
</getCategoryListResponse>

```

# getContactHistoryFilterRequest

This request returns the filter used in getting the contact history. The filter then can be used in a `getItemsRequest` or with the cursor calls. --For GroupWise 2012 and later.

## Request

```
<getContactHistoryFilterRequest>
  <id type="types:uid"/>
</getContactHistoryFilterRequest>
```

## Response

```
<getContactHistoryFilterResponse>
  <filter type="types:Filter"/>
  <status type="types:Status"/>
</getContactHistoryFilterResponse>
```

## Elements

### id

ID of the contact to generate the filter.

### filter

Filter to use in getting the items associated with the contact.

### code

Specifies the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getContactHistoryFilterRequest>
  <id>4625EDFB.domain.P01.104.1776172.1.BC.1@56:
    45080270.domain.P01.104.1776172.1.3.1@53</id>
</getContactHistoryFilterRequest>
```

```
<getContactHistoryFilterResponse>
  <filter>
    <element type="FilterGroup">
      <op>or</op>
      <element type="FilterGroup">
        <op>and</op>
        <element type="FilterGroup">
          <op>or</op>
          <element type="FilterEntry">
            <op>contains</op>
            <field>to</field>
            <value>Preston Stephenson</value>
```

```

</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>cc</field>
  <value>Preston Stephenson</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>bc</field>
  <value>Preston Stephenson</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>to</field>
  <value>pstephenson@prestons.provo.novell.com</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>cc</field>
  <value>pstephenson@prestons.provo.novell.com</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>bc</field>
  <value>pstephenson@prestons.provo.novell.com</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>to</field>
  <value>pstephenson@pstephenson.provo.novell.com</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>cc</field>
  <value>pstephenson@pstephenson.provo.novell.com</value>
</element>
<element type="FilterEntry">
  <op>contains</op>
  <field>bc</field>
  <value>pstephenson@pstephenson.provo.novell.com</value>
</element>
</element>
<element type="FilterGroup">
  <op>or</op>
  <element type="FilterEntry">
    <op>contains</op>
    <field>from</field>
    <value>Preston Stephenson</value>
  </element>
</element>
</element>
<element type="FilterGroup">
  <op>or</op>
  <element type="FilterEntry">
    <op>contains</op>

```

```

        <field>from</field>
        <value>Preston Stephenson</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>author</field>
        <value>Preston Stephenson</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>creator</field>
        <value>Preston Stephenson</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>from</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>author</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>creator</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>from</field>
        <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>author</field>
        <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>creator</field>
        <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
</element>
<element type="FilterGroup">
    <op>or</op>
    <element type="FilterEntry">
        <op>contains</op>
        <field>ItemContact</field>
        <value>.J_H    &quot;f &lt;ar?!    !}&lt;/value>
    </element>
</element>
</filter>
<status>

```

```

    <code>0</code>
  </status>
</getContactHistoryFilterResponse>

<createCursorRequest>
  <container>folders</container>
  <filter>
    <element type="FilterGroup">
      <op>or</op>
      <element type="FilterGroup">
        <op>and</op>
        <element type="FilterGroup">
          <op>or</op>
          <element type="FilterEntry">
            <op>contains</op>
            <field>to</field>
            <value>Preston Stephenson</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>cc</field>
            <value>Preston Stephenson</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>bc</field>
            <value>Preston Stephenson</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>to</field>
            <value>pstephenson@prestons.provo.novell.com</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>cc</field>
            <value>pstephenson@prestons.provo.novell.com</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>bc</field>
            <value>pstephenson@prestons.provo.novell.com</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>to</field>
            <value>pstephenson@pstephenson.provo.novell.com</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>
            <field>cc</field>
            <value>pstephenson@pstephenson.provo.novell.com</value>
          </element>
          <element type="FilterEntry">
            <op>contains</op>

```

```

        <field>bc</field>
        <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>to</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>cc</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>bc</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
</element>
<element type="FilterGroup">
    <op>or</op>
    <element type="FilterEntry">
        <op>contains</op>
        <field>from</field>
        <value>Preston Stephenson</value>
    </element>
</element>
</element>
<element type="FilterGroup">
    <op>or</op>
    <element type="FilterEntry">
        <op>contains</op>
        <field>from</field>
        <value>Preston Stephenson</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>author</field>
        <value>Preston Stephenson</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>creator</field>
        <value>Preston Stephenson</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>from</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
        <op>contains</op>
        <field>author</field>
        <value>pstephenson@prestons.provo.novell.com</value>
    </element>

```

```

    <element type="FilterEntry">
      <op>contains</op>
      <field>creator</field>
      <value>pstephenson@prestons.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
      <op>contains</op>
      <field>from</field>
      <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
      <op>contains</op>
      <field>author</field>
      <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
    <element type="FilterEntry">
      <op>contains</op>
      <field>creator</field>
      <value>pstephenson@pstephenson.provo.novell.com</value>
    </element>
  </element>
  <element type="FilterGroup">
    <op>or</op>
    <element type="FilterEntry">
      <op>contains</op>
      <field>ItemContact</field>
      <value>.J_H    "f &lt;ar?!  !} !</value>
    </element>
  </element>
</filter>
</createCursorRequest>

<createCursorResponse>
  <cursor>1217918620</cursor>
  <status>
    <code>0</code>
  </status>
</createCursorResponse>

<readCursorRequest>
  <container>folders</container>
  <cursor>1217918620</cursor>
  <forward>true</forward>
  <position>current</position>
  <count>0</count>
</readCursorRequest>

<readCursorResponse>
  <items>
    <item type="Task">
      <id>4D385ABA.domain.P01.100.1776172.1.25C6D.1@3:
        A.domain.P01.100.0.1.0.1@19</id>
      <sid>154733</sid>
      <version>7</version>
    </item>
  </items>
</readCursorResponse>

```



```

<modified>2011-01-25T08:00:04Z</modified>
<container sid="5">A.domain.P01.100.0.1.0.1@19</container>
<created>2011-01-20T22:54:34Z</created>
<status>
  <accepted>1</accepted>
  <read>1</read>
</status>
<msgId>4D385ABA.domain.P01.200.20000C2.1.42666.1</msgId>
<clientMessageId>4D38BD2A.290 : 194 : 18298</clientMessageId>
<source>received</source>
<delivered>2011-01-20T22:54:35Z</delivered>
<security>Normal</security>
<subject>my task</subject>
<distribution>
  <from>
    <displayName>Preston Stephenson</displayName>
    <email>pstephenson@prestons.provo.novell.com</email>
    <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
  </from>
  <to>Preston Stephenson</to>
  <sendoptions>
    <statusTracking>All</statusTracking>
    <notifyRecipients>1</notifyRecipients>
  </sendoptions>
</distribution>
<options>
  <priority>Standard</priority>
</options>
<size>939</size>
<checklist>
  <sequence>1073743359</sequence>
  <dueDate>2011-01-22T19:00:00Z</dueDate>
</checklist>
<retentionModified>2011-01-20T22:54:34Z</retentionModified>
<junkMailEvaluation>
  <line>Message is not eligible for Junk Mail handling</line>
  <line>Message is from an internal sender</line>
</junkMailEvaluation>
<junkMailSettings>
  <line>Junk List is enabled</line>
  <line>Junk Mail using personal address books is not enabled</line>
  <line>Junk iCal Mail using personal address books is not enabled</
line>
  <line>Block List is enabled</line>
</junkMailSettings>
<iCalId>2011-01-20T22:54:34Z_477A00C2@prestons.provo.novell.com</
iCalId>
  <startDate>2011-01-25</startDate>
  <dueDate>2011-01-22</dueDate>
  <assignedDate>2011-01-22</assignedDate>
  <taskPriority>0</taskPriority>
  <timezone>
    <daylight>
      <name>Mountain Daylight Time</name>
      <month>3</month>

```

```

        <dayOfWeek occurrence="Second">Sunday</dayOfWeek>
        <hour>0</hour>
        <minute>0</minute>
        <offset>-21600</offset>
    </daylight>
    <standard>
        <name>Mountain Standard Time</name>
        <month>11</month>
        <dayOfWeek occurrence="First">Sunday</dayOfWeek>
        <hour>0</hour>
        <minute>0</minute>
        <offset>-25200</offset>
    </standard>
</timezone>
</item>
<item type="Note">
    <id>4D385A5E.domain.P01.100.1776172.1.25C6A.1@2:
        A.domain.P01.100.0.1.0.1@19</id>
    <sid>154730</sid>
    <version>3</version>
    <modified>2011-01-20T22:53:02Z</modified>
    <container sid="5">A.domain.P01.100.0.1.0.1@19</container>
    <created>2011-01-20T22:53:01Z</created>
    <status>
        <accepted>1</accepted>
        <read>1</read>
    </status>
    <msgId>4D385A5D.domain.P01.200.20000C2.1.42661.1</msgId>
    <clientMessageId>4D38BCCD.130 : 194 : 18298</clientMessageId>
    <source>received</source>
    <delivered>2011-01-20T22:53:03Z</delivered>
    <security>Normal</security>
    <subject>my note</subject>
    <distribution>
        <from>
            <displayName>Preston Stephenson</displayName>
            <email>pstephenson@prestons.provo.novell.com</email>
            <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
        </from>
        <to>Preston Stephenson</to>
        <sendoptions>
            <statusTracking>None</statusTracking>
            <notifyRecipients>1</notifyRecipients>
        </sendoptions>
    </distribution>
    <options>
        <priority>Standard</priority>
    </options>
    <size>939</size>
    <retentionModified>2011-01-20T22:53:02Z</retentionModified>
    <junkMailEvaluation>
        <line>Message is not eligible for Junk Mail handling</line>
        <line>Message is from an internal sender</line>
    </junkMailEvaluation>
    <junkMailSettings>

```

```

        <line>Junk List is enabled</line>
        <line>Junk Mail using personal address books is not enabled</line>
        <line>Junk iCal Mail using personal address books is not enabled</
line>
        <line>Block List is enabled</line>
    </junkMailSettings>
    <iCalId>2011-01-20T22:53:01Z_477A00C2@prestons.provo.novell.com</
iCalId>
    <startDate>2011-01-22</startDate>
    <timezone>
        <daylight>
            <name>Mountain Daylight Time</name>
            <month>3</month>
            <dayOfWeek occurrence="Second">Sunday</dayOfWeek>
            <hour>0</hour>
            <minute>0</minute>
            <offset>-21600</offset>
        </daylight>
        <standard>
            <name>Mountain Standard Time</name>
            <month>11</month>
            <dayOfWeek occurrence="First">Sunday</dayOfWeek>
            <hour>0</hour>
            <minute>0</minute>
            <offset>-25200</offset>
        </standard>
    </timezone>
</item>
<item type="Appointment">
    <id>4D385A03.domain.P01.100.1776172.1.25C67.1@4:
        A.domain.P01.100.0.1.0.1@19</id>
    <sid>154727</sid>
    <version>3</version>
    <modified>2011-01-20T22:51:31Z</modified>
    <container sid="5">A.domain.P01.100.0.1.0.1@19</container>
    <created>2011-01-20T22:51:31Z</created>
    <status>
        <accepted>1</accepted>
        <read>1</read>
    </status>
    <msgId>4D385A03.domain.P01.200.20000C2.1.4265C.1</msgId>
    <clientMessageId>4D38BC73.63A : 194 : 18298</clientMessageId>
    <source>received</source>
    <delivered>2011-01-20T22:51:32Z</delivered>
    <security>Normal</security>
    <subject>my appointment</subject>
    <distribution>
        <from>
            <displayName>Preston Stephenson</displayName>
            <email>pstephenson@prestons.provo.novell.com</email>
            <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
        </from>
        <to>Preston Stephenson</to>
        <sendoptions>
            <statusTracking>All</statusTracking>

```

```

        <notification>
            <accepted>
                <mail>1</mail>
            </accepted>
        </notification>
        <notifyRecipients>1</notifyRecipients>
    </sendoptions>
</distribution>
<options>
    <priority>Standard</priority>
</options>
<size>939</size>
<retentionModified>2011-01-20T22:51:31Z</retentionModified>
<junkMailEvaluation>
    <line>Message is not eligible for Junk Mail handling</line>
    <line>Message is from an internal sender</line>
</junkMailEvaluation>
<junkMailSettings>
    <line>Junk List is enabled</line>
    <line>Junk Mail using personal address books is not enabled</line>
    <line>Junk iCal Mail using personal address books is not enabled</
line>
    <line>Block List is enabled</line>
</junkMailSettings>
    <iCalId>2011-01-20T22:51:31Z_477A00C2@prestons.provo.novell.com</
iCalId>
    <startDate>2011-01-04T22:30:00Z</startDate>
    <endDate>2011-01-04T23:00:00Z</endDate>
    <acceptLevel>Busy</acceptLevel>
    <alarm enabled="1">600</alarm>
    <allDayEvent>0</allDayEvent>
    <place>here</place>
    <timezone>
        <daylight>
            <name>Mountain Daylight Time</name>
            <month>3</month>
            <dayOfWeek occurrence="Second">Sunday</dayOfWeek>
            <hour>0</hour>
            <minute>0</minute>
            <offset>-21600</offset>
        </daylight>
        <standard>
            <name>Mountain Standard Time</name>
            <month>11</month>
            <dayOfWeek occurrence="First">Sunday</dayOfWeek>
            <hour>0</hour>
            <minute>0</minute>
            <offset>-25200</offset>
        </standard>
    </timezone>
</item>
<item type="Mail">
    <id>4D355FEF.domain.P01.100.1776172.1.25C3B.1@1:
        7.domain.P01.100.0.1.0.1@16</id>
    <sid>154683</sid>

```

```

<version>3</version>
<modified>2011-01-18T16:41:16Z</modified>
<container>7.domain.P01.100.0.1.0.1@16</container>
<created>2011-01-18T16:38:48Z</created>
<status>
  <opened>1</opened>
  <read>1</read>
</status>
<msgId>4D355FEF.domain.P01.200.20000C2.1.425A3.1</msgId>
<clientMessageId>4D35C218.CBD : 194 : 18298</clientMessageId>
<source>received</source>
<delivered>2011-01-18T16:40:00Z</delivered>
<security>Normal</security>
<subject>Ð Ð%Ð»Ð´Ð¶Ñ?</subject>
<distribution>
  <from>
    <displayName>Preston Stephenson &lt;ps></displayName>
    <email>pstephenson@prestons.provo.novell.com</email>
    <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
  </from>
  <to>pstephenson@prestons.provo.novell.com</to>
  <cc>pstephenson@gw.novell.com</cc>
  <sendoptions>
    <statusTracking>All</statusTracking>
    <notifyRecipients>1</notifyRecipients>
  </sendoptions>
</distribution>
<options>
  <priority>Standard</priority>
</options>
<size>797</size>
<retentionModified>2011-01-18T16:39:59Z</retentionModified>
<junkMailEvaluation>
  <line>Message is not eligible for Junk Mail handling</line>
  <line>Message is from an internal sender</line>
</junkMailEvaluation>
<junkMailSettings>
  <line>Junk List is enabled</line>
  <line>Junk Mail using personal address books is not enabled</line>
  <line>Junk iCal Mail using personal address books is not enabled</
line>
  <line>Block List is enabled</line>
</junkMailSettings>
</item>
</items>
<status>

```

```
    <code>0</code>
  </status>
</readCursorResponse>

<destroyCursorRequest>
  <container>folders</container>
  <cursor>1217918620</cursor>
</destroyCursorRequest>

<destroyCursorResponse>
  <status>
    <code>0</code>
  </status>
</destroyCursorResponse>
```

# getCustomListRequest

Returns a list of custom field definitions that are defined on a post office. No values for the fields are returned

## Request

```
<getCustomListRequest/>
```

## Response

```
<getCustomListResponse>
  <customs type="types:CustomList" />
  <status type="types:Status" />
</getCustomListResponse>
```

## Elements

### customs

Returns the list of custom fields that are defined in the GroupWise system.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getCustomListRequest/>

<getCustomListResponse>
  <customs>
    <custom>
      <field>customSOAP1</field>
      <value>Custom Field Value</value>
    </custom>
  </customs>
  <status>
    <code>0</code>
  </status>
</getCustomListResponse>
```

# getDeltasRequest

Retrieves changes in the GroupWise Address Book. It returns a list of items that have changed in the GroupWise Address Book since the last call to getDeltasRequest.

## Request

```
<getDeltasRequest">
  <container type="types:uid"/>
  <view type="types:View"/>
  <deltaInfo type="types:DeltaInfo"/>
</getDeltasRequest>
```

## Response

```
<getDeltasResponse>
  <items type="types:Items"/>
  <deltaInfo type="types:DeltaInfo"/>
  <status type="types:Status"/>
</getDeltasResponse>
```

## Elements

### container

Specifies the uid of the GroupWise Address Book. This is the only valid container at this time.

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned.

### deltaInfo

Specifies the current state of the GroupWise Address Book.

### items

Specifies the items that have changed in the GroupWise Address Book since the last synchronization.

### deltaInfo

Specifies the current state of the GroupWise Address Book.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.



## Remarks

getDeltasRequest tracks additions, deletions, and modifications to the GroupWise Address Book, which is a good way to synchronize a local copy of the GroupWise Address Book.

The following are the steps required to retrieve and synchronize the GroupWise Address Book:

1. Get all the items in the GroupWise Address Book.
2. Get the current state of the GroupWise Address Book by calling [getDeltaInfoRequest](#).
3. After a period of time, during which the GroupWise Address Book changed, call getDeltasRequest to retrieve the list of changes, passing the deltaInfo element that was returned in Step 2.

## Example

```
<getDeltasRequest>
  <container>GroupWiseSystemAddressBook@52</container>
  <view/>
  <deltaInfo>
    <firstSequence>23</firstSequence>
    <lastSequence>23</lastSequence>
    <lastTimePORebuild >0</lastTimePORebuild>
  </deltaInfo>
</getDeltasRequest>

<getDeltasResponse>
  <items>
    <item type="Contact">
      <id>EDC03180-01EE-0000-9D71-
        9E00F4007600@56:GroupWiseSystemAddressBook@52</id>
      <name>Admin Admin</name>
      <version>1</version>
      <uuid>EDC03180-01EE-0000-9D71-9E00F4007600</uuid>
      <sync>update</sync>
      <domain>domain1</domain>
      <postOffice>pol</postOffice>
      <distinguishedName>Admin.GW</distinguishedName>
      <userid>Admin</userid>
      <fullName>
        <displayName>Admin Admin</displayName>
        <firstName>Admin</firstName>
        <lastName>Admin</lastName>
      </fullName>
    </item>
  </items>
</getDeltasResponse>
```

```
        <emailList primary="Admin.pol.domain1">
            <email>Admin.pol.domain1</email>
        </emailList>
    </tem>
</items>
<deltaInfo>
    <count>1</count>
    <lastSequence>23</lastSequence>
    <lastTimePORebuild>0</lastTimePORebuild>
</deltaInfo>
<status>
    <code>0</code>
</status>
</getDeltasResponse>
```

# getDeltaInfoRequest

Returns the current state of the GroupWise Address Book. The deltaInfo structure contains pointers or sequence numbers into a dynamic list of changes in the GroupWise Address Book. An application can update its GroupWise Address Book cache based on the sequence numbers returned in the deltaInfo structure

## Request

```
<getDeltaInfoRequest">  
  <container type="types:uid"/>  
  <view type="string"/>  
</getDeltaInfoRequest>
```

## Response

```
<getDeltaInfoResponse">  
  <deltaInfo type="types:DeltaInfo"/>  
  <status type="types:Status"/>  
</getDeltaInfoResponse>
```

## Elements

### container

Specifies the uid of the GroupWise GroupWise Address Book. This is the only valid uid at this time.

### deltaInfo

Specifies the current state of the GroupWise Address Book.

### deltaInfo count

In the request, specifies what items to return. -1 returns all the items in the list. If omitted, all the items are returned. In the response, specifies the actual number of items that were returned.

### deltaInfo firstSequence

In the request, specifies the sequence or position to start the read. In the response, specifies the first valid sequent in the list.

### deltaInfo lastSequence

In the request, not used. In the response, specifies the last sequence number that was successfully read.

### deltaInfo lastTimePORebuild

Specifies the last time the administrator rebuilt the post office. A post office rebuild resets all the sequence numbers. If a rebuild occurs, a synchronization is required before new deltas can be applied to a local list.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

**status**

Returns the success or failure of the method.

**Remarks**

The deltaInfo element has the following elements: count, firstSequence, lastSequence, and lastTimePORebuild. Each element can be used in the request and in the response from the POA.

**Example**

```
<getDeltaInfoRequest>
  <container>GroupWiseSystemAddressBook@52</container>
</getDeltaInfoRequest>

<getDeltaInfoResponse>
  <deltaInfo>
    <firstSequence>23</firstSequence>
    <lastSequence>40</lastSequence>
    <lastTimePORebuild>0</lastTimePORebuild>
  </deltaInfo>
  <status>
    <code>0</code>
  </status>
</getDeltaInfoResponse>
```

# getDiskSpaceUsageRequest

Retrieves the disk space usage for a user.

## Request

```
<getDiskSpaceUsageRequest>
```

## Response

```
<getDiskSpaceUsageRequest>  
  <diskUsage type="types:DiskSpaceUsage" />  
  <status type="types:Status" />  
</getDiskSpaceUsageRequest>
```

## Elements

### diskUsage

Returns the disk usage for the logged-in user.

### status

Returns the success or failure of the method.

# getDocumentTypeListRequest

Returns the list of document types in a library.

## Request

```
<getDocumentTypeListRequest>
  <library/>
</getDocumentTypeListRequest>
```

## Response

```
<getDocumentTypeListResponse>
  <items type="types:DocumentTypeList" />
  <status type="types:Status" />
</getDocumentTypeListResponse>
```

## Elements

### library

Specifies which GroupWise library to query.

### items

Returns the list of document types for a library.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getDocumentTypeListRequest>
  <library>domain1.pol.pol Library@67</library>
</getDocumentTypeListRequest>
```

```
<getDocumentTypeListResponse>
  <items>
    <item type="DocumentType">
      <name>Agenda</name>
      <life>99</life>
      <maximumversions>100</maximumversions>
      <ageAction>archive</ageAction>
    </item>
    <item type="DocumentType">
      <name>Contract</name>
      <life>99</life>
```

```
        <maximumversions>100</maximumversions>
        <ageAction>archive</ageAction>
    </item>
    <item type="DocumentType">
        <name>Corresp</name>
        <life>99</life>
        <maximumversions>100</maximumversions>
        <ageAction>archive</ageAction>
    </item>
</items>
<status>
    <code>0</code>
</status>
</getDocumentTypeListResponse>
```

# getFolderRequest

Returns the specified folder. The folder can be specified by passing in the folder ID that was returned from [getFolderListRequest](#) or by passing the folder type.

## Request

```
<getFolderRequest>
  <id type="types:uid"/>
  <folderType type="types:FolderType"/>
  <types type="string"/>
  <source type="string"/>
  <view type="string"/>
</getFolderRequest>
```

## Response

```
<getFolderResponse>
  <folder type="types:Folder">
    <status type="types:Status">
</getFolderResponse>
```

## Elements

### id

Specifies the ID of the folder.

### folderType

Specifies an alternative way to specify the folder. The Junk Mail folder cannot be selected by folderType. It must be selected with the folder ID.

### types

Specifies the type of items to return. The types element is used to narrow the count or unreadCount on a folder. For example, if you want the count or unreadCount for appointments and tasks, types should equal “appointments tasks.”

### source

Specifies the source of items to return. The source element is used to narrow the count or unreadCount on a folder. For example, if you want the count or unreadCount for received and draft items, source should equal “received draft.”

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned. Valid views are as follows:

count	Retrieves the count of items in a folder. The count is available only on mailbox, personal, and draft folders. Hidden items in a folder are not part of the count.
unreadCount	Retrieves the count of unread items in a folder.
default	Retrieves other folder fields.



**folder**

Specifies the selected folder details.

**code**

Returns the error number related to the event. 0 indicates that the request was successful.

**status**

Returns the success or failure of the method.

**Example**

```
<getFolderRequest>
  <folderType>Mailbox</folderType>
  <view>default count unreadCount</view>
</getFolderRequest>

<getFolderResponse>
  <folder>
    <id>7.domain1.pol.100.0.1.0.1@16</id>
    <name>Mailbox</name>
    <modified>2012-08-25T17:38:39Z</modified>
    <parent>6.domain1.pol.100.0.1.0.1@15</parent>
    <count>5</count>
    <hasUnread>1</hasUnread>
    <unreadCount>3</unreadCount>
    <sequence>5</sequence>
    <isSystemFolder>1</isSystemFolder>
    <folderType>Mailbox</folderType>
  </folder>
  <status>
    <code>0</code>
  </status>
</getFolderResponse>
```

# getFolderListRequest

Returns the list of folders.

## Request

```
<getFolderListRequest>
  <parent type="types:uid"/>
  <view type="string"/>
  <recurse type="boolean"/>
  <imap type="boolean"/>
  <nntp type="boolean"/>
</getFolderListRequest>
```

## Response

```
<getFolderListResponse>
  <folders type="types:FolderList"/>
  <status type="types:Status"/>
</getFolderListResponse>
```

## Elements

### parent

Specifies the ID of the folder from which to get the list of folders. If the value of the parent property is the string folders, the root folder is searched.

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned.

### recurse

Specifies whether subfolders beneath the parent folder are returned.

### imap

Specifies if IMAP folders are returned.

### nntp

Specifies if NNTP folders are returned.

### folders

Returns the list of folders.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getFolderListRequest>
  <parent>folders</parent>
  <view/>
  <recurse>true</recurse>
  <imap>false</imap>
  <nntp>false</nntp>
</getFolderListRequest>

<getFolderListResponse>
  <folders>
    <folder type="SystemFolder">
      <id>6.domain1.pol.100.0.1.0.1@15</id>
      <name>ul</name>
      <modified>2005-06-29T23:28:26Z</modified>
      <count>1</count>
      <hasUnread>1</hasUnread>
      <unreadCount>1</unreadCount>
      <sequence>0</sequence>
      <isSystemFolder>1</isSystemFolder>
      <folderType>Root</folderType>
    </folder>
    <folder type="SystemFolder">
      <id>7.domain1.pol.100.0.1.0.1@16</id>
      <name>Mailbox</name>
      <modified>2012-08-25T17:38:39Z</modified>
      <parent>6.domain1.pol.100.0.1.0.1@15</parent>
      <count>2</count>
      <sequence>5</sequence>
      <isSystemFolder>1</isSystemFolder>
      <folderType>Mailbox</folderType>
      <acl>
        <entry>
          <displayName>u2</displayName>
          <email>u2.pol.domain1</email>
          <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
          <rights>
            <read>1</read>
            <add>1</add>
            <edit>1</edit>
            <delete>1</delete>
          </rights>
        </entry>
      </acl>
      <isSharedByMe>1</isSharedByMe>
    </folder>
    <folder type="SystemFolder">
      <id>19.domain1.pol.100.0.1.0.1@30</id>
      <name>Sent Items</name>
      <modified>2012-09-05T20:43:47Z</modified>
      <parent>6.domain1.pol.100.0.1.0.1@15</parent>
      <sequence>6</sequence>
      <isSystemFolder>1</isSystemFolder>
      <folderType>SentItems</folderType>
```

```

</folder>
<folder type="SystemFolder">
  <id>A.domain1.pol.100.0.1.0.1@19</id>
  <name>Calendar</name>
  <modified>2012-09-06T23:08:05Z</modified>
  <parent>6.domain1.pol.100.0.1.0.1@15</parent>
  <count>7</count>
  <sequence>7</sequence>
  <isSystemFolder>1</isSystemFolder>
  <folderType>Calendar</folderType>
  <acl>
    <entry>
      <displayName>u2</displayName>
      <email>u2.pol.domain1</email>
      <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
      <rights>
        <read>1</read>
        <add>1</add>
      </rights>
    </entry>
  </acl>
  <isSharedByMe>1</isSharedByMe>
</folder>
<folder type="SharedFolder">
  <id>44FF0075.domain1.pol.100.16E3837.1.FB4.1@35</id>
  <name>u1 Personal Calendar</name>
  <modified>2012-09-06T23:10:37Z</modified>
  <parent>A.domain1.pol.100.0.1.0.1@19</parent>
  <count>1</count>
  <sequence>0</sequence>
  <calendarAttribute>
    <flags>ShowInList</flags>
  </calendarAttribute>
  <acl>
    <entry>
      <displayName>u2</displayName>
      <email>u2.pol.domain1@phantom.com</email>
      <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
      <rights>
        <read>1</read>
        <add>1</add>
        <edit>1</edit>
        <delete>1</delete>
      </rights>
      <status>accepted</status>
    </entry>
  </acl>
  <isSharedByMe>1</isSharedByMe>
</folder>
<folder type="ContactFolder">
  <id>1B.domain1.pol.100.0.1.0.1@32</id>
  <name>Contacts</name>
  <modified>2012-08-25T17:38:39Z</modified>
  <parent>6.domain1.pol.100.0.1.0.1@15</parent>
  <sequence>8</sequence>

```

```

    <isSystemFolder>1</isSystemFolder>
    <folderType>Contacts</folderType>
    <addressBook>42C510EA.domain1.pol.104.16E3837.1.1.1@53
    </addressBook>
  </folder>
  <folder type="SystemFolder">
    <id>10.domain1.pol.100.0.1.0.1@23</id>
    <name>Documents</name>
    <modified>2012-08-25T17:38:39Z</modified>
    <parent>6.domain1.pol.100.0.1.0.1@15</parent>
    <sequence>9</sequence>
    <isSystemFolder>1</isSystemFolder>
    <folderType>Documents</folderType>
  </folder>
  <folder type="QueryFolder">
    <id>11.domain1.pol.100.0.1.0.1@14</id>
    <name>Authored</name>
    <modified>2005-06-29T23:28:15Z</modified>
    <parent>10.domain1.pol.100.0.1.0.1@23</parent>
    <sequence>0</sequence>
    <folderType>Query</folderType>
  </folder>
  <folder type="SystemFolder">
    <id>1A.domain1.pol.100.0.1.0.1@31</id>
    <name>Checklist</name>
    <modified>2012-08-25T17:38:39Z</modified>
    <parent>6.domain1.pol.100.0.1.0.1@15</parent>
    <sequence>10</sequence>
    <isSystemFolder>1</isSystemFolder>
    <folderType>Checklist</folderType>
  </folder>
  <folder type="SystemFolder">
    <id>D.domain1.pol.100.0.1.0.1@22</id>
    <name>Work In Progress</name>
    <modified>2012-08-25T17:38:39Z</modified>
    <parent>6.domain1.pol.100.0.1.0.1@15</parent>
    <sequence>11</sequence>
    <isSystemFolder>1</isSystemFolder>
    <folderType>Draft</folderType>
  </folder>
  <folder type="SystemFolder">
    <id>C.domain1.pol.100.0.1.0.1@21</id>
    <name>Cabinet</name>
    <modified>2012-08-25T17:38:39Z</modified>
    <parent>6.domain1.pol.100.0.1.0.1@15</parent>
    <count>3</count>
    <hasUnread>1</hasUnread>
    <unreadCount>3</unreadCount>
    <sequence>12</sequence>
    <isSystemFolder>1</isSystemFolder>
    <folderType>Cabinet</folderType>
  </folder>
  <folder type="SystemFolder">
    <id>1C.domain1.pol.100.0.1.0.1@33</id>
    <name>Junk Mail</name>

```

```

        <modified>2012-08-25T17:38:39Z</modified>
        <parent>6.domain1.pol.100.0.1.0.1@15</parent>
        <sequence>13</sequence>
        <isSystemFolder>1</isSystemFolder>
        <folderType>JunkMail</folderType>
    </folder>
    <folder type="SystemFolder">
        <id>9.domain1.pol.100.0.1.0.1@18</id>
        <name>Trash</name>
        <modified>2012-08-25T17:38:39Z</modified>
        <parent>6.domain1.pol.100.0.1.0.1@15</parent>
        <count>20</count>
        <hasUnread>1</hasUnread>
        <unreadCount>21</unreadCount>
        <sequence>14</sequence>
        <isSystemFolder>1</isSystemFolder>
        <folderType>Trash</folderType>
    </folder>
</folders>
<status>
    <code>0</code>
</status>
</getFolderListResponse>

```

# getFreeBusyRequest

Returns free/busy information for GroupWise users. This method is called with [startFreeBusySessionRequest](#) and [closeFreeBusySessionRequest](#) and can be called more than once to retrieve the free/busy information for all users defined in [startFreeBusySessionRequest](#).

## Request

```
<getFreeBusyRequest>
  <freeBusySessionId>
  <view type="string" />
</getFreeBusyRequest>
```

## Response

```
<getFreeBusyResponse>
  <freeBusyStats type="types:FreeBusyStats" />
  <freeBusyInfo type="types:FreeBusyInfoList" />
  <status type="types:Status" />
</getFreeBusyResponse>
```

## Elements

### freeBusySessionId

Specifies the session ID, as returned in the [startFreeBusySessionResponse](#). [startFreeBusySessionRequest](#) must be called before [getFreeBusyRequest](#).

### view

Can pass "from" or "place".

### freeBusyStats

Returns the current state of the free/busy search on the GroupWise POA.

### responded

Specifies the number of users in the free/busy request that have responded with free/busy information.

### outstanding

Specifies the number of users in the free/busy request that have not yet responded with free/busy information.

### total

Specifies the total number of users in the free/busy request.

### freeBusyInfo

Specifies the free/busy blocks for users.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

## status

Returns the success or failure of the method.

## Example

```
<getFreeBusyRequest>
  <freeBusySessionId>354</freeBusySessionId>
</getFreeBusyRequest>

<getFreeBusyResponse>
  <freeBusyInfo>
    <freeBusyStats>
      <responded>2</responded>
      <outstanding>0</outstanding>
      <total>2</total>
    </freeBusyStats>
    <user>
      <email>user1@dmz4.provo.novell.com</email>
      <blocks>
        <block>
          <startDate>2012-03-19T16:00:00Z</startDate>
          <endDate>2012-03-19T16:30:00Z</endDate>
          <acceptLevel>Busy</acceptLevel>
          <subject>A tea party at MadHatter's place</subject>
        </block>
        <block>
          <startDate>2004-03-19T20:00:00Z</startDate>
          <endDate>2004-03-19T20:30:00Z</endDate>
          <acceptLevel>Busy</acceptLevel>
          <subject>Watch the match (stadium)</subject>
        </block>
        <block>
          <startDate>2004-03-19T20:00:00Z</startDate>
          <endDate>2004-03-19T20:30:00Z</endDate>
          <acceptLevel>Busy</acceptLevel>
          <subject>watch the match </subject>
        </block>
        <block>
          <startDate>2004-03-19T07:00:00Z</startDate>
          <endDate>2004-03-19T15:00:00Z</endDate>
          <acceptLevel>OutOfOffice</acceptLevel>
        </block>
        <block>
          <startDate>2004-03-20T00:00:00Z</startDate>
          <endDate>2004-03-22T15:00:00Z</endDate>
          <acceptLevel>OutOfOffice</acceptLevel>
        </block>
        <block>
          <startDate>2004-03-23T00:00:00Z</startDate>
          <endDate>2004-03-23T15:00:00Z</endDate>
          <acceptLevel>OutOfOffice</acceptLevel>
        </block>
      </blocks>
    </user>
```



```
<user>
  <email>user2@dmz4.provo.novell.com</email>
  <blocks>
    <block>
      <startDate>2004-03-19T07:00:00Z</startDate>
      <endDate>2004-03-19T15:00:00Z</endDate>
      <acceptLevel>OutOfOffice</acceptLevel>
    </block>
    <block>
      <startDate>2004-04-08T23:00:00Z</startDate>
      <endDate>2004-04-09T14:00:00Z</endDate>
      <acceptLevel>OutOfOffice</acceptLevel>
    </block>
  </blocks>
</user>
</freeBusyInfo>
<status>
  <code>0</code>
</status>
</getFreeBusyResponse>
```

# getItemRequest

Returns the item specified by the ID.

## Request

```
<getItemRequest>
  <id type="types:uid">
    <view type="string"/>
  </getItemRequest>
```

## Response

```
<getItemResponse>
  <item type="types:Item"/>
  <status type="types:Status"/>
</getItemResponse>
```

### id

Specifies the ID of the item to return.

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned.

### item

Returns the specified item.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

This method is used to get the following item types:

- ♦ Mail
- ♦ Appointments
- ♦ Notes
- ♦ Tasks
- ♦ Phone
- ♦ Contacts
- ♦ Organizations
- ♦ Resources
- ♦ Groups
- ♦ Documents

## Example

```
<getItemRequest>
  <id>45013330.domain1.pol.100.16E3837.1.FBB.1@1:7.
    domain1.pol.100.0.1.0.1@16</id>
  <view>default recipients message/RTF attachments recipientStatus
    peek</view>
</getItemRequest>

<getItemResponse>
  <item type="Mail">
    <id>45013330.domain1.pol.100.16E3837.1.FBB.1@1:7.
      domain1.pol.100.0.1.0.1@16</id>
    <version>1</version>
    <modified>2012-09-08T15:09:04Z</modified>
    <container>7.domain1.pol.100.0.1.0.1@16</container>
    <created>2012-09-08T15:09:04Z</created>
    <customs>
      <custom>
        <field>customSOAP1</field>
        <value>Test Value</value>
      </custom>
    </customs>
    <status>
      <accepted>1</accepted>
      <opened>1</opened>
      <read>1</read>
    </status>
    <msgId>45013330.domain1.pol.100.16E3837.1.FBB.1</msgId>
    <source>personal</source>
    <delivered>2012-09-08T15:09:04Z</delivered>
    <security>Normal</security>
    <subject>Setting Custom fields</subject>
    <distribution>
      <from>
        <displayName>ul</displayName>
        <email>ul@phantom.com</email>
        <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
      </from>
      <sendoptions>
        <statusTracking>None</statusTracking>
      </sendoptions>
    </distribution>
    <options>
      <priority>Standard</priority>
    </options>
    <size>256</size>
  </item>
  <status>
    <code>0</code>
  </status>
</getItemResponse>
```

# getItemsRequest

Returns a set of items in a container (folders, address books, or libraries). There are three ways to specify the items to return: specify the container in the container's element, specify the item IDs in the items element, or specify a filter.

## Request

```
<getItemsRequest>
<container type="types:uid" />
<view type="string"/>
<filter type="types:Filter" />
<items type="types:ItemRefList" />
<count type="int" />
```

## Response

```
<getItemsResponse>
<items type="types:Items" />
<sids type="string" />
<status type="types:Status" />
```

## Elements

### container

(optional) Specifies the folder, address book, or library container ID. If the container is not specified, the filter is used. If the filter is not present, an error is returned.

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned.

### filter

Specifies the items to return based on a filter.

### items

Specifies a list of item ids to return.

### sids

If "sids" is supplied in the view, the items are returned in the as a space delimited string of sid (short ID's) values.

GroupWise 2012 and later.

### count

Specifies the number of items to return. If the count element is -1, all items are returned.

### items

Returns the specified items.

**code**

Specifies the error number related to the event. 0 indicates that the request was successful.

**status**

Returns the success or failure of the method.

**Remarks**

This method is used to get the following item types:

- ♦ Folders
- ♦ Query Folders
- ♦ Shared Folders
- ♦ GroupWise Address Book
- ♦ Personal Address Book
- ♦ Documents
- ♦ Libraries

If a container is not supplied, either a filter or an item reference list is required to limit the search. Also, if the container is not specified, all folders are searched.

The `getItemsRequest` reads all the items in the specified container. The number of items returned can be reduced by providing the count element. For example, if a container has 4,000 items, `getItemsRequest` returns all 4,000 items if a count is not specified. If a count of 200 is specified, 4,000 items are read and only 200 items are returned. Reading all the items in a container and returning a subset of items is not efficient.

Be aware that `getItemsRequest` returns only 5,000 items. If there are more than 5,000 items available, an error message is returned, directing you to refine your search criteria.

You can use filters to limit your searches.

**Example**

This section contains several examples on retrieving items.

## Get Folder Items

```
<getItemsRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item type="Mail">
      <id>45003893.domain1.pol.100.16E3837.1.FBA.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <version>3</version>
      <modified>2012-09-07T22:44:53Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <created>2012-09-07T21:19:47Z</created>
      <status>
        <opened>1</opened>
        <read>1</read>
      </status>
      <msgId>45003893.domain1.pol.200.20000CB.1.1532.1</msgId>
      <messageId><45003893.2618.00CB.0@phantom.com></messageId>
      <source>received</source>
      <delivered>2012-09-07T21:19:47Z</delivered>
      <security>Normal</security>
      <subject>Attachment attached</subject>
      <distribution>
        <from>
          <displayName>ul</displayName>
          <email>ul@phantom.com</email>
          <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
        </from>
        <to>ul</to>
        <sendoptions>
          <statusTracking>None</statusTracking>
        </sendoptions>
      </distribution>
      <options>
        <priority>Standard</priority>
      </options>
      <hasAttachment>1</hasAttachment>
      <size>1801</size>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>
```

## Library

```
<getItemsRequest>
  <container>domain1.pol.pol Library@67</container>
  <view/>
  <filter/>
  <count>0</count>
</getItemsRequest>

<getItemsResponse>
  <items>
    <item type="Document">
      <id>domain1.pol.pol Library:5@68</id>
      <version>5</version>
      <container>domain1.pol.pol Library@67</container>
      <created>2012-09-13T10:45:49Z</created>
      <subject>This is a zip file that contains a
        PALM Dictionary</subject>
      <library>
        <displayName>pol Library</displayName>
        <email>pol Library@phantom.com</email>
      </library>
      <documentNumber>5</documentNumber>
      <documentTypeName>Document</documentTypeName>
      <author>
        <displayName>ul</displayName>
        <email>ul@phantom.com</email>
        <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
      </author>
      <creator>
        <displayName>ul</displayName>
        <email>ul@phantom.com</email>
        <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
      </creator>
      <officialversion>2</officialversion>
      <currentversion>2</currentversion>
      <current>
        <entry>
          <displayName>Creator Access</displayName>
          <email>C</email>
          <rights>
            <read>1</read>
            <edit>1</edit>
            <delete>1</delete>
            <share>1</share>
            <manage>1</manage>
          </rights>
        </entry>
      </current>
    </official>
    <entry>
      <displayName>Creator Access</displayName>
      <email>C</email>
      <rights>
        <read>1</read>
```

```

        <edit>1</edit>
        <delete>1</delete>
        <share>1</share>
        <manage>1</manage>
    </rights>
</entry>
</official>
<other>
    <entry>
        <displayName>Creator Access</displayName>
        <email>C</email>
        <rights>
            <read>1</read>
            <edit>1</edit>
            <delete>1</delete>
            <share>1</share>
            <manage>1</manage>
        </rights>
    </entry>
</other>
</item>
</items>
<status>
    <code>0</code>
</status>
</getItemsResponse>

```

## Filter

The following filtered request retrieves tasks in the specified container:

```

<getItemsRequest>
    <container>A.AutoDomain.AutoPO1.100.0.1.0.1@19</container>
    <filter>
        <element>
            <op>eq</op>
            <field>@type</field>
            <value>Task</value>
        </element>
    </filter>
</getItemsRequest>

```

The following filtered request retrieves personal or received items in the specified container:



```

<getItemsRequest>
  <container>A.AutoDomain.AutoPO1.100.0.1.0.1@19</container>
  <filter>
    <elements>
      <op>or</op>
      <element>
        <op>eq</op>
        <field>source</field>
        <value>received</value>
      </element>
      <element>
        <op>eq</op>
        <field>source</field>
        <value>personal</value>
      </element>
    </elements>
  </filter>
</getItemsRequest>

```

The following filtered request retrieves items in the specified container that were received after 5:00pm UTC on September 12, 2012:

```

<getItemsRequest>
  <container>A.AutoDomain.AutoPO1.100.0.1.0.1@19</container>
  <filter>
    <elements>
      <op>and</op>
      <element>
        <op>gte</op>
        <field>startDate</field>
        <value>2012-09-12T17:00:00Z</value>
      </element>
      <element>
        <op>eq</op>
        <field>source</field>
        <value>received</value>
      </element>
    </element>
  </filter>
</getItemsRequest>

```

# getJunkEntriesRequest

Returns the junk, block, and trust list entries for a user.

## Request

```
<getJunkEntriesRequest>  
  <container>  
</getJunkEntriesRequest>
```

## Response

```
<getJunkEntriesResponse>  
  <junk>  
  <block>  
  <trust>  
  <status>  
</getJunkEntriesResponse>
```

## Elements

### container

Specifies the junk, block, or trust list to retrieve. If the container is null, the junk, block, and trust lists are returned.

### junk

Specifies items to move to the junk folder if the sender's email address or domain matches any of the returned entries.

### block

Specifies items not to deliver if the sender's email address or domain matches any of the returned entries.

### trust

Specifies items to deliver if the sender's email address or domain matches any of the returned entries.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getJunkEntriesRequest/>

<getJunkEntriesResponse>
  <junk>
    <entry>
      <id>44FDACF0.domain1.pol.104.16E3837.1.103.1@66</id>
      <match>user@junk.com</match>
      <matchType>email</matchType>
      <listType>junk</listType>
      <lastUsed>2012-09-05T22:59:28Z</lastUsed>
      <version>1</version>
      <modified>2012-09-05T22:59:28Z</modified>
    </entry>
  </junk>
  <block>
    <entry>
      <id>44FDACF0.domain1.pol.104.16E3837.1.104.1@66</id>
      <match>user@block.com</match>
      <matchType>email</matchType>
      <listType>block</listType>
      <lastUsed>2012-09-05T22:59:28Z</lastUsed>
      <version>1</version>
      <modified>2012-09-05T22:59:28Z</modified>
    </entry>
  </block>
  <trust>
    <entry>
      <id>44FDACF0.domain1.pol.104.16E3837.1.105.1@66</id>
      <match>user@trust.com</match>
      <matchType>email</matchType>
      <listType>trust</listType>
      <lastUsed>2012-09-05T22:59:28Z</lastUsed>
      <version>1</version>
      <modified>2012-09-05T22:59:28Z</modified>
    </entry>
  </trust>
  <status>
    <code>0</code>
  </status>
</getJunkEntriesResponse>
```

# getJunkMailSettingsRequest

Returns the junk mail settings for a user.

## Request

```
<getJunkMailSettingsRequest/>
```

## Response

```
<getJunkMailSettingsResponse>
  <settings type="types:SettingsList">
    <status type="types:Status"/>
  </getJunkMailSettingsResponse>
```

## Elements

### settings

Returns the junk mail handling settings that were defined by the user.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getJunkMailSettingsRequest/>

<getJunkMailSettingsResponse>
  <settings>
    <setting>
      <field>useBlockList</field>
      <value>1</value>
    </setting>
    <setting>
      <field>useJunkList</field>
      <value>1</value>
    </setting>
    <setting>
      <field>usePAB</field>
      <value>0</value>
    </setting>
    <setting>
      <field>persistence</field>
      <value>14</value>
    </setting>
  </settings>
  <status type="types:Status"/>
</getJunkMailSettingsResponse>
```

```
</setting>
<setting>
  <field>autoDelete</field>
  <value>0</value>
</setting>
<setting>
  <field>iCalUsePAB</field>
  <value>0</value>
</setting>
</settings>
<status>
  <code>0</code>
</status>
</getJunkMailSettingsResponse>
```

# getLibraryItemRequest

Returns the document and version from the specified library. The library ID and document number are required. The version number is optional.

## Request

```
<getLibraryItemRequest>
  <library type="types:uid"/>
  <documentNumber type="unsignedInt"/>
  <versionNumber type="string"/>
</getLibraryItemRequest>
```

## Response

```
<getLibraryItemResponse>
  <item type="types:Item"/>
  <status type="types:Status"/>
</getLibraryItemResponse>
```

## Elements

### library

Specifies the library ID for the document and version.

### documentNumber

Specifies the the document number.

### versionNumber

Specifies the the version number. To get a version of a document, the document number is required:

- ♦ current
- ♦ official
- ♦ a specific version number

### item

Specifies the document details.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getLibraryItemRequest>
  <library>domain1.pol.pol Library@67</library>
  <documentNumber>5</documentNumber>
  <versionNumber>current</versionNumber>
</getLibraryItemRequest>

<getLibraryItemResponse>
  <item type="version">
    <id>domain1.pol.pol Library:5:1@69</id>
    <version>3</version>
    <container>domain1.pol.pol Library:5@67</container>
    <created>2012-09-13T10:45:49Z</created>
    <library>
      <displayName>pol Library</displayName>
      <email>pol Library@phantom.com</email>
    </library>
    <documentNumber>5</documentNumber>
    <versionCreator>
      <displayName>ul</displayName>
      <email>ul@phantom.com</email>
      <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
    </versionCreator>
    <retrievedBy>
      <displayName>ul</displayName>
      <email>ul@phantom.com</email>
      <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
    </retrievedBy>
    <retrievedDate>2012-09-13T10:45:50Z</retrievedDate>
    <versionNumber>1</versionNumber>
    <versionDescription>Original</versionDescription>
    <versionStatus>available</versionStatus>
    <life>0</life>
    <ageAction>archive</ageAction>
    <fileSize>2212871</fileSize>
    <filename>zip</filename>
  </item>
  <status>
    <code>0</code>
  </status>
</getLibraryItemResponse>
```

# getLibraryListRequest

Returns the GroupWise libraries that are defined on the system.

## Request

```
<getLibraryListRequest/>
```

## Response

```
<getLibraryListResponse>
  <libraries type="types:LibraryList"/>
  <status type="types:Status"/>
</getLibraryListResponse>
```

## Elements

### libraries

Specifies the libraries on the GroupWise system.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getLibraryListRequest/>

<getLibraryListResponse>
  <libraries>
    <library type="Library">
      <id>domain1.pol.pol Library@67</id>
      <name>pol Library</name>
      <domain>domain1</domain>
      <postOffice>pol</postOffice>
    </library>
  </libraries>
  <status>
    <code>0</code>
  </status>
</getLibraryListResponse>
```



# getMemberOfRequest

Returns the list of distribution lists (GroupWise Address Book groups) that has the particular user. This is used by WebAccess to determine if a user is part of a group when the group is used in an Access Control List.

## Request

```
<getMemberOfRequest>
  <userid type="xs:string" minOccurs="0" />
  <postOffice type="xs:string" minOccurs="0" />
  <domain type="xs:string" minOccurs="0" />
</getMemberOfRequest>
```

## Response

```
<getMemberOfResponse>
  <group type="tns:Owner" minOccurs="0" maxOccurs="unbounded" />
  <status type="types:Status"/>
</getMemberOfResponse>
```

## Elements

### userid

Userid to match.

### postOffice

Post office to match.

### domain

Domain to match.

### group

0 or more group identifiers.

### code

Returns the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getMemberOfRequest>
  <userid>pstephenson</userid>
  <postOffice>pol</postOffice>
  <domain>domain</domain>
</getMemberOfRequest>
```

```
<getMemberOfResponse>
  <group>
    <domain>domain</domain>
    <postOffice>P01</postOffice>
    <userid>all</userid>
  </group>
  <group>
    <domain>domain</domain>
    <postOffice>P01</postOffice>
    <userid>dlP01</userid>
  </group>
  <group>
    <domain>domain</domain>
    <postOffice>P01</postOffice>
    <userid>P01Group</userid>
  </group>
  <status>
    <code>0</code>
  </status>
</getMemberOfResponse>
```

# getNotifyListRequest

Get a list of users in the notification list. See [createNotifyRequest](#). --For GroupWise 8.0 SP1 and later.

## Request

```
<getNotifyListRequest/>
```

## Response

```
<getNotifyListResponse>
  <list type="types:NotifyList"/>
  <status type="types:Status"/>
</getNotifyListResponse>
```

## Elements

### list

List of NotifyEntry elements.

### code

Specifies the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getNotifyListRequest/>

<getNotifyListResponse>
  <list>
    <entry>
      <displayName>Craig Johnson</displayName>
      <email>cjohnson.P01.domain@prestons.provo.novell.com</email>
      <uuid>AD712880Â„0B55Â„0000Â„B976Â„8E90E5B002BD</uuid>
      <id>48CE5C4E.domain.P01.100.1776172.1.2C86.1@78</id>
      <rights>
        <alarm>1</alarm>
        <notification>1</notification>
      </rights>
    </entry>
```

```
<entry>
  <displayName>Preston Stephenson</displayName>
  <email>pstephenson.P01.domain@prestons.provo.novell.com</email>
  <id>45080216.domain.P01.100.1776172.1.22.1@78</id>
  <rights>
    <alarm>1</alarm>
    <notification>1</notification>
  </rights>
</entry>
</list>
<status>
  <code>0</code>
</status>
</getNotifyListResponse>
```

# getProxyAccessListRequest

Returns the list of users that can proxy or access the current GroupWise account. It also returns the rights the users have to the account.

## Request

```
<getProxyAccessListRequest />
```

## Response

```
<getProxyAccessListResponse>
  <accessRights type="types:AccessRightList" />
  <status type="types:Status" />
</getProxyAccessListResponse>
```

## Elements

### accessRights

Specifies a list of users and their rights to the user's account.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getProxyAccessListRequest />

<getProxyAccessListResponse>
  <accessRights>
    <entry>
      <displayName>ul</displayName>
      <email>ul.pol.domain1@phantom.com</email>
      <uuid>1DEB6970175B0000890F6E00D5004E00</uuid>
      <id>1DEB6970175B0000890F6E00D5004E00@60</id>
      <appointment>
        <read>1</read>
      </appointment>
      <mail>
        <read>1</read>
```

```
        <write>1</write>
    </mail>
    <note>
        <read>1</read>
    </note>
    <task>
        <read>1</read>
    </task>
</entry>
</accessRights>
<status>
    <code>0</code>
</status>
</getProxyAccessListResponse>
```

# getProxyListRequest

Returns the list of users that have proxy rights to the current account.

## Request

```
<getProxyListRequest/>
```

## Response

```
<getProxyListResponse>  
  <proxies type="types:ProxyList" />  
  <status type="types:Status" />  
</getProxyListResponse>
```

## Elements

### proxies

Specifies the list of users that can proxy this account.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

The proxy list is managed by the user. A user's proxy list is not automatically updated when a user gives or removes access to another user. A user is only added to the proxy list after a successful proxy. For example, when userA successfully proxies into userB's account, userB is added to the proxy list for userA. The proxy list is really a history of user accounts that have successfully been proxied.

A user is removed from the proxy list only if they are specifically deleted. For example, if userC gives proxy rights to userA, userC is not returned in getProxyListResponse until userA has successfully proxied into userC's account. If userC is already in the proxy list for userA and userC removes the rights for userA, userA still sees userC in his or her proxy list. If userA tries to proxy userC, access is denied. UserA still needs to delete userC from the proxy list.

## Example

```
<getProxyListRequest/>

<getProxyListResponse>
  <proxies>
    <proxy>
      <displayName>ul</displayName>
      <email>ul</email>
      <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
      <id>1DEB6970-175B-0000-890F-6E00D5004E00@38</id>
    </proxy>
  </proxies>
  <status>
    <code>0</code>
  </status>
</getProxyListResponse>
```



# getQuickMessagesRequest

Provides quicker access to new or modified item or to all of the items than the standard `getItems` methods. Before implementing `getQuickMessagesRequest`, consider using [GroupWise Events](#) instead. GroupWise Events is superior and is the recommended way to learn about changes to a GroupWise user's mailbox.

## Request

```
<getQuickMessagesRequest>
  <list type="types:MessageList"/>
  <ref="types:startDate"/>
  <container type="types:uid"/>
  <types type="string"/>
  <source type="string"/>
  <view type="string"/>
  <count type="int"/>
</getQuickMessagesRequest>
```

## Response

```
<getQuickMessagesResponse>
  <startDate/>
  <items type="types:Items"/>
  <status type="types:Status"/>
</getQuickMessagesResponse>
```

## Elements

### list

Specifies what items to query:

---

new	Provides the fastest access to new items. The new list supports a view that can return the whole item. The larger the view, the slower this approach becomes.
modified	Provides slower access to modified items.
all	Provides the slowest access to all items.

---

### startDate

Returns items that are newer or equal to the `startDate`.

### container

Specifies the container to query. If a container is not provided, the query is across all folders

### types

Specifies the message types to return. If message type is not provided, it returns all message types. If the container is the calendar and the message type is not provided, calendar items are returned. Calendar items are appointments, notes, and tasks.

**source**

Specifies the source of the items. If the source is not provided, draft, personal, and received items are returned.

**view**

Specifies the fields to be returned. If the view is not included and the list is not equal to new, all the indexed fields are returned. If the list is new, you can specify a view to return the entire item.

**count**

Specifies the number of items to return. If no count is given, all items are returned.

**startDate**

Specifies the time value just before the last query. You can then pass in startDate to getQuickMessages subsequently, which ensures that the client is always working with the server time when queries are performed. It also mitigates the time difference between the client and server.

**items**

Returns the items that match the query.

**code**

Returns the error number related to the event. 0 indicates that the request was successful.

**status**

Returns the success or failure of the method.

**Remarks**

getQuickMessagesRequest provides quicker access because it accesses only those fields that are stored in the GroupWise database indexes. Some fields that are stored in the indexes and returned are categories, container, id, message type, modified, originalSubject, sendOptions, source, startDate, status, and subject.

getQuickMessagesRequest has the following limitations:

- ♦ Is not supported in proxy mode.
- ♦ Cannot be used on contact, query, and trash folders.
- ♦ Does not return a response when items are deleting.

## Example

```
<getQuickMessagesRequest>
  <list>New</list>
  <container>6.domain1.pol.100.0.1.0.1@15</container>
  <view>default recipients message</view>
</getQuickMessagesRequest>

<getQuickMessagesResponse>
  <startDate>2012-09-13T19:10:52Z</startDate>
  <items>
    <item type="Mail">
      <id>44EDB77E.domain1.pol.100.16E3837.1.F7A.1@1:6.
        domain1.pol.100.0.1.0.1@15</id>
      <version>6</version>
      <modified>2012-09-13T19:10:50Z</modified>
      <container>6.domain1.pol.100.0.1.0.1@15</container>
      <created>2012-08-24T20:28:14Z</created>
      <status>
        <opened>1</opened>
        <read>1</read>
      </status>
      <msgId>44EDB77E.domain1.pol.200.20000CB.1.14B0.1</msgId>
      <messageId><44EDB77E.2618.00CB.0@phantom.com></messageId>
      <source>received</source>
      <delivered>2012-08-24T20:28:14Z</delivered>
      <security>Normal</security>
      <subject>new</subject>
      <distribution>
        <from>
          <displayName>ul</displayName>
          <email>ul@phantom.com</email>
          <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
        </from>
        <to>ul</to>
        <recipients>
          <recipient>
            <displayName>ul</displayName>
            <email>ul@phantom.com</email>
            <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
            <distType>TO</distType>
            <recipType>User</recipType>
          </recipient>
        </recipients>
        <sendoptions>
          <statusTracking>None</statusTracking>
        </sendoptions>
      </distribution>
    </item>
  </items>
</getQuickMessagesResponse>
```

```
        </distribution>
        <message>
            <part contentType="text/plain" length="8">YXNkZg==</part>
        </message>
        <options>
            <priority>Standard</priority>
        </options>
        <size>991</size>
    </item>
</items>
<status>
    <code>0</code>
</status>
</getQuickMessagesResponse>
```

# getRuleListRequest

Returns the list of rules for a user.

## Request

```
<getRuleListRequest/>
```

## Response

```
<getRuleListResponse>
  <rules type="types:RuleList"/>
  <status type="types:Status"/>
</getRuleListResponse>
```

## Elements

### rules

Specifies the rules for a user's account.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getRuleListRequest/>
```

```
<getRuleListResponse>
  <rules>
    <rule type="Rule">
      <id>44630E88.domain1.po1.100.16E3837.1.B5E.1@10</id>
      <name>Rule 1</name>
      <version>12</version>
      <modified>2012-09-13T19:17:22Z</modified>
      <execution>New</execution>
      <sequence>0</sequence>
      <enabled>1</enabled>
      <types>Appointment Mail Task</types>
      <source>received sent</source>
      <actions>
        <action>
          <type>Reply</type>
          <item type="Mail">
            <id>4506D010.domain1.po1.200.20000CB.1.1548.1@1</id>
            <version>1</version>
            <modified>2012-09-12T21:19:44Z</modified>
            <created>2012-09-12T21:19:42Z</created>
            <delivered>2012-09-12T21:19:42Z</delivered>
```

```

        <security>Normal</security>
        <subject>Test Subject</subject>
        <distribution>
            <sendoptions>
                <statusTracking>None</statusTracking>
            </sendoptions>
        </distribution>
        <options>
            <priority>Standard</priority>
        </options>
        <size>256</size>
    </item>
</action>
</actions>
</rule>
<rule type="Rule">
    <id>4411628C.domain1.pol.100.16E3837.1.AA7.1@10</id>
    <name>Vacation Rule</name>
    <version>9</version>
    <modified>2012-05-24T04:59:11Z</modified>
    <execution>New</execution>
    <sequence>-1</sequence>
    <source>received</source>
    <actions>
        <action>
            <type>Unknown</type>
            <item type="Mail">
                <id>441162AB.domain1.pol.200.20000CB.1.1228.1@1</id>
                <version>1</version>
                <modified>2012-03-10T18:27:39Z</modified>
                <created>2012-03-10T18:27:39Z</created>
                <delivered>2012-03-10T18:27:39Z</delivered>
                <security>Normal</security>
                <subject>Vacation Rule from ul</subject>
                <distribution>
                    <from>
                        <displayName>ul</displayName>
                        <email>ul@phantom.com</email>
                        <uuid>1DEB6970-175B-0000-890F-zz

```

```

        6E00D5004E00</uuid>
      </from>
      <sendoptions>
        <statusTracking>None</statusTracking>
      </sendoptions>
    </distribution>
    <message>
      <part contentType="text/plain" length="8">
        YXNkZg==</part>
      </message>
      <options>
        <priority>Standard</priority>
      </options>
      <size>272</size>
    </item>
  </action>
</actions>
</rule>
</rules>
<status>
  <code>0</code>
</status>
</getRuleListResponse>

```

# getSettingsRequest

Returns all user settings. The `getSettingsResponse` provides groups of settings. The groups are similar to the Windows client groups

## Request

```
<getSettingsRequest>
  <id" type="string"/>
</getSettingsRequest>
```

## Response

```
<getSettingsResponse>
  <settings type="types:Settings"/>
  <status type="types:Status"/>
</getSettingsResponse>
```

## Elements

### id

Specifies the name of the setting. If blank, all settings are returned. Can contain the name of a group of settings or of an individual setting.

### settings

Specifies the settings for a user. Each setting has a field and a value. It also has a locked element to signify that the setting cannot be changed because the administrator has locked down the setting.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

You can get all user settings, a group of settings, or an individual setting. If you want to retrieve all the settings for a user, don't add a value for the ID. If you want to retrieve a group of settings for a user, specify the group name in the setting. For example, to get all the settings in the Environment Settings, specify `EnvironmentSettings` for the ID. If you want to retrieve an individual setting, specify the name of the setting in the ID.

Be aware that the GroupWise clients hide some of the complexities of client settings. Make sure you understand the consequences of changing a setting before calling `getSettingsRequest`.



## Example

```
<getSettingsRequest/>

<getSettingsResponse>
  <settings>
    <group type="EnvironmentSettings">
      <setting>
        <field>allowSharedFolders</field>
        <value>1</value>
        <locked>1</locked>
      </setting>
      <setting>
        <field>settingsLanguage</field>
        <value>US</value>
      </setting>
      <setting>
        <field>allowSharedAddressBooks</field>
        <value>1</value>
        <locked>1</locked>
      </setting>
      <setting>
        <field>allowPOP_IMAPAccounts</field>
        <value>1</value>
        <locked>1</locked>
      </setting>
      <setting>
        <field>allowNNTPAccounts</field>
        <value>1</value>
        <locked>1</locked>
      </setting>
      <setting>
        <field>showIMPresence</field>
        <value>1</value>
      </setting>
    </group>
    <group type="TimezoneSettings">
      <setting>
        <field>daylightTime</field>
        <value>1</value>
      </setting>
      <setting>
        <field>daylightBeginDate</field>
        <value>0000-04-02</value>
      </setting>
      <setting>
        <field>daylightEndDate</field>
        <value>0000-10-29</value>
      </setting>
      <setting>
        <field>daylightChange</field>
        <value>01:00</value>
      </setting>
      <setting>
        <field>daylightHemisphere</field>
```

```

        <value>Northern</value>
    </setting>
    <setting>
        <field>GMTOffset</field>
        <value>-7:00</value>
    </setting>
    <setting>
        <field>daylightFormula</field>
        <value>00000105</value>
    </setting>
</group>
<group type="MailMessageSettings">
    <setting>
        <field>mailPriority</field>
        <value>Standard</value>
    </setting>
    <setting>
        <field>notifyRecipient</field>
        <value>1</value>
    </setting>
    <setting>
        <field>mailReturnOpen</field>
    </setting>
    <setting>
        <field>mailReturnCompleted</field>
    </setting>
    <setting>
        <field>mailReturnDelete</field>
    </setting>
    <setting>
        <field>mailStatusInfo</field>
        <value>DeliveredAndOpened</value>
    </setting>
    <setting>
        <field>mailReplyRequested</field>
        <value>None</value>
    </setting>
    <setting>
        <field>mailExpireDays</field>
        <value>0</value>
    </setting>
    <setting>
        <field>mailAutoDelete</field>
        <value>0</value>
    </setting>
</group>
<group type="AppointmentMessageSettings">
    <setting>
        <field>appointmentPriority</field>
        <value>Standard</value>
    </setting>
    <setting>
        <field>appointmentReturnOpen</field>
    </setting>
    <setting>

```

```

        <field>appointmentReturnAccept</field>
    </setting>
    <setting>
        <field>appointmentReturnDelete</field>
    </setting>
    <setting>
        <field>appointmentStatusInfo</field>
        <value>Full</value>
    </setting>
    <setting>
        <field>appointmentDefaultLength</field>
        <value>01:00</value>
    </setting>
</group>
<group type="TaskMessageSettings">
    <setting>
        <field>taskPriority</field>
        <value>Standard</value>
    </setting>
    <setting>
        <field>taskReturnOpen</field>
    </setting>
    <setting>
        <field>taskReturnAccept</field>
    </setting>
    <setting>
        <field>taskReturnCompleted</field>
    </setting>
    <setting>
        <field>taskReturnDelete</field>
    </setting>
    <setting>
        <field>taskStatusInfo</field>
        <value>Full</value>
    </setting>
</group>
<group type="NoteMessageSettings">
    <setting>
        <field>notePriority</field>
        <value>Standard</value>
    </setting>
    <setting>
        <field>noteReturnOpen</field>
    </setting>
    <setting>
        <field>noteReturnDelete</field>
    </setting>
    <setting>
        <field>noteStatusInfo</field>
        <value>DeliveredAndOpened</value>
    </setting>
    <setting>
        <field>noteReturnAccept</field>
    </setting>
</group>

```

```

<group type="AdvancedSettings">
  <setting>
    <field>sendSecurity</field>
    <value>Normal</value>
  </setting>
  <setting>
    <field>concealedSubject</field>
    <value>0</value>
  </setting>
  <setting>
    <field>delayDelivery</field>
    <value>-1</value>
  </setting>
  <setting>
    <field>outboxInsert</field>
    <value>1</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>itemConversions</field>
    <value>0</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>routePasswordRequired</field>
    <value>0</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>asteriskSendRestriction</field>
    <value>postOffice</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>autoSpellCheck</field>
    <value>0</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>addSignature</field>
    <value>3</value>
  </setting>
  <setting>
    <field>encryptMessages</field>
    <value>0</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>disallowSMIME</field>
    <value>0</value>
    <locked>1</locked>
  </setting>
  <setting>
    <field>internetStatusTrackingSettings</field>
  </setting>

```

```

    <setting>
      <field>simpleForward</field>
      <value>0</value>
    </setting>
    <setting>
      <field>allowRuleReplyMoreThanOnce</field>
      <value>0</value>
    </setting>
  </group>
  <group type="BusySettings">
    <setting>
      <field>busyStartTime</field>
      <value>08:00</value>
    </setting>
    <setting>
      <field>busyEndTime</field>
      <value>17:00</value>
    </setting>
    <setting>
      <field>busyInterval</field>
      <value>15</value>
    </setting>
    <setting>
      <field>busyDays</field>
      <value>Monday,Tuesday,Wednesday,Thursday,Friday</value>
    </setting>
    <setting>
      <field>busyExtended</field>
      <value>0</value>
    </setting>
    <setting>
      <field>busySearchRange</field>
      <value>7</value>
    </setting>
  </group>
  <group type="CalendarViewSettings">
    <setting>
      <field>appointmentIncludeSelf</field>
      <value>1</value>
    </setting>
    <setting>
      <field>startOfWorkday</field>
      <value>08:00</value>
    </setting>
    <setting>
      <field>endOfWorkday</field>
      <value>17:00</value>
    </setting>
    <setting>
      <field>workdays</field>
      <value>Monday,Tuesday,Wednesday,Thursday,Friday</value>
    </setting>
  </group>
  <group type="DiscardSettings">
    <setting>

```

```

        <field>mailDelete</field>
        <value>manualDeleteArchive</value>
    </setting>
    <setting>
        <field>appointmentDelete</field>
        <value>manualDeleteArchive</value>
    </setting>
    <setting>
        <field>trashPurge</field>
        <value>autoPurgeAfterTrashDays</value>
    </setting>
    <setting>
        <field>trashDays</field>
        <value>7</value>
    </setting>
    <setting>
        <field>messageSendLimit</field>
        <value>0</value>
        <locked>1</locked>
    </setting>
    <setting>
        <field>boxThresholdLimit</field>
        <value>0</value>
        <locked>1</locked>
    </setting>
    <setting>
        <field>boxSizeLimit</field>
        <value>0</value>
        <locked>1</locked>
    </setting>
    <setting>
        <field>enableBoxThresholdNotificaion</field>
        <value>0</value>
    </setting>
    <setting>
        <field>enableBoxSizeNotificaion</field>
        <value>0</value>
    </setting>
</group>
<group type="AppointmentViewSettings">
    <setting>
        <field>appointmentAlarmSet</field>
        <value>1</value>
    </setting>
    <setting>
        <field>appointmentAlarmMinutes</field>
        <value>5</value>
    </setting>
    <setting>
        <field>allDayEventAlarmSet</field>
        <value>0</value>
    </setting>
    <setting>
        <field>allDayEventAlarmMinutes</field>
        <value>720</value>
    </setting>

```

```

        </setting>
    </group>
    <group type="SPAMSettings">
        <setting>
            <field>useBlockList</field>
            <value>1</value>
        </setting>
        <setting>
            <field>useJunkList</field>
            <value>1</value>
        </setting>
        <setting>
            <field>usePAB</field>
            <value>0</value>
        </setting>
        <setting>
            <field>persistence</field>
            <value>14</value>
        </setting>
        <setting>
            <field>autoDelete</field>
            <value>0</value>
        </setting>
        <setting>
            <field>iCalUsePAB</field>
            <value>0</value>
        </setting>
    </group>
</settings>
<status>
    <code>0</code>
</status>
</getSettingsResponse>

```

# getSignaturesRequest

Returns all the user's signatures. HTML signatures are stored in a user's database and returned in MIME format.

## Request

```
<getSignaturesRequest>
  <global type="boolean"/>
</getSignaturesRequest>
```

## Response

```
<getSignaturesResponse>
  <signatures" type="types:Signatures"/>
  <status" type="types:Status"/>
</getSignaturesResponse>
```

## Elements

### global

Is not used at this time. There is no way to retrieve the global signature using Web Services.

### signatures

Returns the user's list of signatures.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.



## Example

```
<getSignaturesRequest/>

<getSignaturesResponse>
  <signatures>
    <signature>
      <id>44FE9BAA.domain1.pol.100.16E3837.1.F95.1@63</id>
      <name>Signature 1</name>
      <default>1</default>
      <part>
        <data>TWltZS1WZXJzaW9uOiAxLjANCkNvbhRlbnQtVHlwZTogbXVsdG1
          wYXJ0L2lpeGVkOyBib3VuZGFyeT0iXl9fXl1lRWE5QVk9KUEhBTk
          ZEVVhMSVlTXl9fXyINCg0KDQotLV9fXl9ZUVhOUF
          ...
          DVUtQQlhRSERLQ0NaTV9fXl8tLQ0KDQotLV9fXl9ZUVhOUF
          ZPSlBIQU5GRFVYTElZU19fXl8tLQ0K</data>
        <size>1888</size>
      </part>
      <global>0</global>
    </signature>
  </signatures>
  <status>
    <code>0</code>
  </status>
</getSignaturesResponse>
```

# getStringRequest

Returns the GroupWise string in the specified language.

## Request

```
<getStringRequest>
  <id type="xs:unsignedInt"/>
  <language type="language"/>
</getStringRequest>
```

## Response

```
<getStringResponse>
  <value type="string"/>
  <status type="types:Status"/>
</getStringResponse>
```

## Elements

### id

The id of the GroupWise string. For example, you can pass an in an id of “35110”. This maps to the SVR\_UNKNOWN\_HOST error string (for example, Error 8926hex in engerr.txt file located in the directory errorCodes in this NDK).

### language

Specifies the return string language.

# getTimestampRequest

Returns the date the user's database was last backedup and the last time retention software processed items in an account. Typically, backup or retention software time-stamps an account when it completes. This method can also be used as a noop method. If the noop element is True, no work is done in the POA to return data. This operation can be used to keep a connection alive.

## Request

```
<getTimestampRequest>
  <backup type="boolean"/>
  <retention type="boolean"/>
  <noop type="boolean"/>
  <retentionModified type="boolean"/>
</getTimestampRequest>
```

## Response

```
<getTimestampResponse>
  <backup type="dateTime"/>
  <retention type="dateTime"/>
  <status type="types:Status"/>
  <retentionModified type="dateTime"/>
</getTimestampResponse>
```

## Elements

### backup

Specifies whether the last backup date on the user's account is returned.

### retention

Specifies whether the last retention date on the user's account is returned.

### noop

Specifies that getTimestampRequest is used to ping the server and keep the connection alive.

### backup

Specifies the date the user's account was last backed up.

### retention

Returns the time stamp that was marked by retention software.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## **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 timestamp. Suppose an item has a retention timestamp. If a user modifies the item by adding a personal attachment or some other significant change, retention software will skip the item because it has already been retained. RetentionModified is now used to catch an item after the first retention and a significant change has occurred to the item.

## **Example**

```
<getTimestampRequest>
  <backup>true</backup>
  <retention>true</retention>
  <noop>false</noop>
</getTimestampRequest>

<getTimestampResponse>
  <backup>2012-07-19T01:00:00Z</backup>
  <retention>2012-08-20T03:00:00Z</retention>
  <getTimestampResponse>
    <status>
      <code>0</code>
    </status>
  </getTimestampResponse>
```

# getTimezoneListRequest

Returns the list of time zone definitions at the post office.

## Request

```
<getTimezoneListRequest/>
```

## Response

```
<getTimezoneListResponse>
  <timezones type="TimezoneList"/>
  <status type="types:Status"/>
</getTimezoneListResponse>
```

## Elements

### timezones

Specifies the list of time zone definitions at the post office.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<getTimezoneListRequest/>

<getTimezoneListResponse>
  <timezones>
    ...
    <timezone>
      <id>DLT</id>
      <description>(GMT-12:00) International Date Line
        West</description>
      <standard>
        <name>Dateline Standard Time</name>
        <offset>-43200</offset>
      </standard>
    </timezone>
    <timezone>
      <id>NZT</id>
      <description>(GMT+12:00) Auckland, Wellington</description>
      <daylight>
        <name>New Zealand Daylight Time</name>
        <month>10</month>
        <dayOfWeek occurrence="First">Sunday</dayOfWeek>
        <hour>2</hour>
        <minute>0</minute>
```

```
        <offset>46800</offset>
    </daylight>
    <standard>
        <name>New Zealand Standard Time</name>
        <month>3</month>
        <dayOfWeek occurrence="Third">Sunday</dayOfWeek>
        <hour>2</hour>
        <minute>0</minute>
        <offset>43200</offset>
    </standard>
</timezone>
</timezones>
<status>
    <code>0</code>
</status>
</getTimezoneListResponse>
```

# getUnreadRequest

WebAccess needed a way to get the list of unread items in the user's mailbox. The list needs to be returned as quickly as possible. This is a specialized call to return that list. There are a few parameters to qualify the list returned. The list returned is a list of sid's (short identifiers). Once you have the list you can pass the sid's in a getItemRequest to return part or all of the items. --For GroupWise 2012 and later.

## Request

```
<getUnreadRequest>
  <startDate type="dateTime" minOccurs="0"/>
  <types type="string" minOccurs="0"/>
  <source type="xs:string" minOccurs="0"/>
</getUnreadRequest>
```

## Response

```
<getUnreadResponse>
  <sids type="xs:string" minOccurs="0"/>
  <status type="types:Status"/>
</getUnreadResponse>
```

## Elements

### startDate

If specified, return items with a startDate greater than or equal to the date. (For Mail, PhoneMessage or DocumentReference, it compares the value against the created date.)

### types

A space delimited string of item types to include in the search. You can have one or more of the following values

- ♦ Mail
- ♦ Appointment
- ♦ Note
- ♦ Task
- ♦ PhoneMessage
- ♦ DocumentReference
- ♦ CalendarItem

(This includes appointments, notes and tasks.)

If you don't supply a value for "types", appointments, mail, notes, tasks and phone messages are returned.

### source

A space delimited string of box types to include in the search. You can have one or more of the following values:

- ♦ received

- ♦ sent
- ♦ personal (posted)
- ♦ draft

If you do not supply a value, received, personal and draft items is returned.

#### sids

The return value is a space-delimited list of sids of the matching items matching the query.

#### status

The query returns approximately 4000 items. If there are more items available, a D11B (53531) WPF\_WARN\_TOO\_MANY\_RECORDS error is returned. To get more items, you must adjust the startDate value and call getUnreadRequest again.

## Example

Here is some sample code. It gets the unread items. If there are too many items, it reads the last item, finds the startDate, and uses that to read the next set of sids.

---

**NOTE:** You might have some duplicate sids in this case. You must merge the sets.

---

```
public void testFindUnread() {
    Calendar start = null;
    GetUnreadResponse uresp = null;
    GetItemsResponse resp = null;
    int i = 0;
    ItemRefList list = null;
    String read;
    String str;
    String view = "id sid status subject startDate created";
    String [] items = null;
    String [] one = {" "};
    StringTokenizer tokens;

    try {
        uresp = m_main.getService().getUnreadRequest( null, "Mail
CalendarItem",
            null, m_main.getSessionId(), m_main.getTrace() );
        if ( null == uresp.getStatus() ) {
            return;
        }

        if ( !(0 == uresp.getStatus().getCode() ||
            0xd11b == uresp.getStatus().getCode() ) ) {
            m_main.displayError( uresp.getStatus(), "testFindUnread" );
            return;
        }
        tokens = new StringTokenizer( uresp.getSids(), " " );
        m_main.displayString( "Count = " +
String.valueOf(tokens.countTokens())
        );
        if ( 0xd11b == uresp.getStatus().getCode() ) {
            items = new String[ tokens.countTokens() ];
```



```

        while ( tokens.hasMoreTokens() ) {
            one[ 0 ] = items[ i++ ] = tokens.nextToken();
        }
        list = new ItemRefList( one );
        resp = m_main.getService().getItemsRequest( "folders", view, null,
            list, 0, m_main.getSessionId(), m_main.getTrace() );
        start = ((Mail)(resp.getItems().getItem()[0])).getCreated();
        uresp = m_main.getService().getUnreadRequest( start,
            "Mail CalendarItem", null, m_main.getSessionId(),
m_main.getTrace()
            );
        tokens = new StringTokenizer( uresp.getSids(), " " );
        m_main.displayString( "Count = " +
            String.valueOf(tokens.countTokens()) );
    }
} catch ( Exception ex ) {
    ex.printStackTrace();
}
}

SOAP Trace
<getUnreadRequest>
  <types>Mail CalendarItem</types>
</getUnreadRequest>

<getUnreadResponse>
  <sids>260815 264110 270837 ... 283851 283848 283845 283842 283839</sids>
  <status>
    <code>53531</code>
    <description>More than 5000 records</description>
  </status>
</getUnreadResponse>

<getItemsRequest>
  <container>folders</container>
  <view>id sid status subject startDate created</view>
  <items>
    <item >283839</item>
  </items>
</getItemsRequest>

```

```

<getItemsResponse>
  <items>
    <item type="Mail">
      <id>4E92DD25.domain.P01.100.1776172.1.454BF.1@1:4B82436A.
        domain.P01.100.1776172.1.E062.1@13</id>
      <sid>283839</sid>
      <container sid="57442">4B82436A.domain.P01.100.1776172.1.E062.1@13
        </container>
      <created>2011-10-10T17:55:17Z</created>
      <source>received</source>
      <delivered>2011-10-10T17:55:17Z</delivered>
      <subject>test three [962]</subject>
    </item>
  </items>
  <status>
    <code>0</code>
  </status>
</getItemsResponse>

<getUnreadRequest>
  <startDate>2011-10-10T17:55:17.000+00:00</startDate>
  <types>Mail CalendarItem</types>
</getUnreadRequest>

<getUnreadResponse>
  <sids>283851 283848 283845 ... 287055 287056</sids>
  <status>
    <code>0</code>
  </status>
</getUnreadResponse>

```

# getUserListRequest

Returns all the users on a GroupWise post office.

## Request

```
<getUserListRequest>
  <name type="string"/>
  <key type="string"/>
  <noop type="boolean"/>
</getUserListRequest>
```

## Response

```
<getUserListResponse>
  <users type="types:UserList"/>
  <status" type="types:Status"/>
</getUserListReponse>
```

## Elements

### name

Specifies the name of the trusted application.

### key

Specifies the value of the trusted application.

### noop

Used to test the connection to the POA without having to be logged in as a user.

**--For GroupWise 8.0 SP1 and later.**

### users

Specifies the list of users on a post office. Each user in the list has a name, email address, unique identifier (uuid), user ID, and its recipient type.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

This method is available only to applications that have logged in as a trusted application. Those with user or proxy authentication will be denied.

Trusted applications are set up by an administrator. A GroupWise administrator provides the trusted application name and key to an application.

If an application has already logged in as a trusted application, the name and key elements are not required. Otherwise, the name and key elements are required.

## Example

```
<getUserListRequest>
  <name>GWWS</name>
  <key>C0C0A3D206A90000875068005A00E700C0C0A3D306A90000875068
    005A00E700</key>
</getUserListRequest>

<getUserListResponse>
  <users>
    <user>
      <name>Resoucel</name>
      <email>Resoucel.pol.domain1</email>
      <uuid>EC1AA110-083F-0000-8109-D1000000AC00</uuid>
      <userid>Resoucel</userid>
      <recipType>Resource</recipType>
    </user>
    <user>
      <name>ul</name>
      <email>ul.pol.domain1</email>
      <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
      <userid>ul</userid>
      <recipType>User</recipType>
    </user>
    ...
  </users>
  <status>
    <code>0</code>
  </status>
</getUserListResponse>
```

# loginRequest

Is used to authenticate to a POA. It can also be used to log in as a user, a proxy user, or as a trusted application user. Keep in mind that login credentials are sent over the wire as plain text. We recommend using SSL to encrypt the data over the wire.

## Request

```
<loginRequest>
  <auth type="types:Authentication"/>
  <language type="language"/>
  <version type="decimal"/>
  <application type="string"/>
  <userid type="xs:boolean"/>
  <system type="boolean"/>
</loginRequest>
```

## Response

```
<loginResponse>
  <session type="string"/>
  <userinfo type="types:UserInfo"/>
  <entry" type="types:AccessRightEntry"/>
  <gwversion" type="string"/>
  <build" type="string"/>
  <redirectToHost" type="types:Host"/>
  <serverUTCtime" type="dateTime"/>
  <status" type="types:Status"/>
</loginResponse>
```

## Elements

### auth

Specifies the type of authentication: PlainText, Proxy or TrustedApplication.

Plain Text	User authentication.
Proxy	Proxy authentication.
TrustedApplication	Trusted application authentication.

### language

Used to return the language specific folder names on system folders. Also error descriptions will be based on the language entered.

### version

Used to control which version of the schema to honor. The current values are:

- ♦ "1.02" - 8.0.0
- ♦ "1.03" - 8.0.1

- ♦ “1.04” - 8.0.2
- ♦ “1.05” - 2012

#### **application**

Specifies a string that describes the application. Administrators can use an HTTP monitor on a POA to view the application that is connected to the POA.

#### **userid**

Specifies the userid, postOffice, and domain information. To retrieve this information, specify TRUE.

#### **session**

Specifies and links the user with the login instance on the POA. The session string is required in all subsequent SOAP methods.

#### **userinfo**

Specifies information about the logged-in user. The userinfo element is returned when PlainText (user) authentication is used.

#### **entry**

Specifies information about the logged-in user. The userinfo element is returned when PlainText (user) or TrustedApplication authentication is used.

#### **gwversion**

Specifies the version of the POA. For example, 8.0.1 is GroupWise 8.0 SP 1.

#### **build**

Specifies the build number of the POA.

#### **redirectToHost**

Specifies a user's POA IP address and port. This element is returned when the specified user is not on the POA. The application should try to log in to the IP address and port contained in the redirectToHost element.

#### **serverUTCTime**

Specifies the date and time of the POA in UTC time.

#### **code**

Returns the error number related to the event. 0 indicates that the request was successful.

#### **status**

Returns the success or failure of the method.

#### **system**

Specifies whether to return the GroupWise system name in the loginResponse.

## Remarks

Before sending any credentials over the wire, verify that your application is talking to a valid GroupWise POA by using the following steps:

- ♦ Examine the server certificate and make sure it is valid.
- ♦ Ensure that the server's certificate was issued by a trusted certificate authority or that it is listed in the trusted certificates for the client.
- ♦ Check the certificate start and expiration dates.

After a successful login, a session string is returned from GroupWise. All subsequent calls to GroupWise Web Services need to include the session string.

Proxy login is a two-step process. First, the main user logs into his or her account (login). Second, the main user proxies into the proxy account (proxy login).

Proxy rights are determined when the proxy login occurs. If the proxy rights are changed after a proxy login, the new proxy rights are not reflected until a logout and another login. For example, user1 proxies into user2's account. User1 has read and write rights to user2's account. While user1 is proxied into user2's account, user2 changes the rights of user1 to read-only. User1 does not see the modified rights until he or she logs out and proxies back into user2's account.

Applications cannot log in to GroupWise resources. To access resources data, log in as the owner of the resource and then proxy into the resource.

## Example

The following sections contain several examples of loginRequest.

### LoginRequest

```
<loginRequest>
  <types:auth type="types:PlainText">
    <types:username>ul</types:username>
    <types:password>ul</types:password>
  </types:auth>
  <application>BEEPClient</application>
</loginRequest>

<loginResponse>
  <session>5cCSFUxJ75UpLxFR</session>
  <userinfo>
    <name>ul</name>
    <email>ul@phantom.com</email>
    <uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
  </userinfo>
  <gwversion>7.0.1</gwversion>
  <build>341</build>
  <serverUTCTime>2012-09-14T18:30:47Z</serverUTCTime>
  <status>
    <code>0</code>
  </status>
</loginResponse>
```

## Proxy Login Request

```
<loginRequest>
<types:auth type="types:Proxy">
<types:username>u1</types:username>
<types:password>u1</types:password>
<types:proxy>u2.pol.domain1</types:proxy>
</types:auth>
<application>BEEPClient</application>
</loginRequest>
<loginResponse>
<session>Tic8EqjFARMR2APt</session>
<entry>
<displayName>u2</displayName>
<email>u2@phantom.com</email>
<uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
<appointment>
<read>1</read>
</appointment>
<mail>
<read>1</read>
<write>1</write>
</mail>
<note>
<read>1</read>
</note>
<task>
<read>1</read>
</task>
</entry>
<gwversion>7.0.1</gwversion>
<build>341</build>
<serverUTCTime>2012-09-14T18:33:51Z</serverUTCTime>
<status>
<code>0</code>
</status>
</loginResponse>
```



## TrustedApplication LoginRequest

```
<loginRequest>
<types:auth type="types:TrustedApplication">
<types:username>ul</types:username>
<types:name>GWWS</types:name>
<types:key>C0C0A3D206A90000875068005A00E700C0C0A3D306A90000875068005A00E70
0</types:key>
</types:auth>
<application>BEEPClient</application>
</loginRequest>
<loginResponse>
<session>ENT_TpWqn8byRcl2</session>
<userinfo>
<name>ul</name>
<email>ul@phantom.com</email>
<uuid>1DEB6970-175B-0000-890F-6E00D5004E00</uuid>
</userinfo>
<gwversion>7.0.1</gwversion>
<build>341</build>
<serverUTCTime>2012-09-14T18:36:34Z</serverUTCTime>
<status>
<code>0</code>
</status>
</loginResponse>
```

# logoutRequest

Logs out the current user.

## Request

```
<logoutRequest />
```

## Response

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

## Elements

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<logoutRequest />  
  
<logoutResponse>  
  <status>  
    <code>0</code>  
  </status>  
</logoutResponse>
```

# markPrivateRequest

Marks the selected item as private. Marking an item private indicates that proxy users do not have view rights to the item.

## Request

```
<markPrivateRequest>
  <items type="types:ItemRefList" />
</markPrivateRequest>
```

## Response

```
<markPrivateResponse>
  <status type="types:Status" />
</markPrivateResponse>
```

## Elements

### items

Specifies the items to mark as private.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<markPrivateRequest>
  <items>
    <item>45057C7F.domain1.pol.100.16E3837.1.FC1.1@1:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
</markPrivateRequest>

<markPrivateResponse>
  <status>
    <code>0</code>
  </status>
</markPrivateResponse>
```

# markReadRequest

Marks the selected items as read.

## Request

```
<markReadRequest>
  <items type="types:ItemRefList" />
</markReadRequest>
```

## Response

```
<markReadResponse>
  <status type="types:Status" />
</markReadResponse>
```

## Elements

### items

Specifies the items to be marked as read.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<markReadRequest>
  <items>
    <item>45057C7F.domain1.pol.100.16E3837.1.FC1.1@1:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
</markReadRequest>

<markReadResponse>
  <status>
    <code>0</code>
  </status>
</markReadResponse>
```

# markUnPrivateRequest

Marks items as unprivate. Marking an item unprivate indicates that proxy users have read rights to the item.

## Request

```
<markUnPrivateRequest>
  <items type="types:ItemRefList" />
</markUnPrivateRequest>
```

## Response

```
<markUnPrivateResponse>
  <status type="types:Status" />
</markUnPrivateResponse>
```

## Elements

### items

Specifies the items to be marked as unprivate.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<markUnPrivateRequest>
  <items>
    <item>45057C7F.domain1.pol.100.16E3837.1.FC1.1@1:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
</markUnPrivateRequest>

<markUnPrivateResponse>
  <status>
    <code>0</code>
  </status>
</markUnPrivateResponse>
```

# markUnReadRequest

Marks the selected items as not read.

## Request

```
<markUnReadRequest>
  <items type="types:ItemRefList" />
</markUnReadRequest>
```

## Response

```
<markUnReadResponse>
  <status type="types:Status" />
</markUnReadResponse>
```

## Elements

### items

Specifies the items to be marked as unread.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<markUnReadRequest>
  <items>
    <item>45057C7F.domain1.pol.100.16E3837.1.FC1.1@1:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
</markUnReadRequest>

<markUnReadResponse>
  <status>
    <code>0</code>
  </status>
</markUnReadResponse>
```

# modifyItemRequest

Modifies the selected item with the given updates. The modifyItemRequest is used to delete, add, and update fields on existing items. Deletions are processed before the additions and updates.

## Request

```
<modifyItemRequest>
  <id type="types:uid"/>
  <notification type="types:SharedFolderNotification"/>
  <updates type="types:ItemChanges"/>
  <recurrenceAllInstances type="unsignedInt"/>
</modifyItemRequest>
```

## Response

```
<modifyItemResponse>
  <modified type="dateTime"/>
  <id type="types:uid"/>
  <status type="types:Status"/>
</modifyItemResponse>
```

## Elements

### id

Specifies the ID of the item to modify.

### notification

Specifies the shared folder notification on a modification.

### updates

Specifies additions, deletions, or modifications to the item.

### recurrenceAllInstances

Specifies the recurrenceKey to apply the modification to all the recurring items. Modifications to distributed items are limited.

### modified

Specifies the time the item was modified.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

The `modifyItemRequest` modifies the following item types:

- ♦ Address Book Items
  - ♦ Contacts
  - ♦ Groups
  - ♦ Organizations
  - ♦ Personal Address Books
  - ♦ Resources
- ♦ Categories
- ♦ Folders
- ♦ Rules
- ♦ Shared Folders
- ♦ Mail Items
  - ♦ Mail
  - ♦ Appointments
  - ♦ Tasks
  - ♦ Notes
  - ♦ Phone

Most fields on a personal or posted mail item can be modified. The only elements that can be modified on a distributed mail item are the subject, categories, and custom fields.

## Example

The following examples show different uses of `modifyItemRequest`.



## Modify a Posted Item

```
<modifyItemRequest>
  <id>45097153.domain1.pol.100.16E3837.1.FC7.1@1:7.
    domain1.pol.100.0.1.0.1@16</id>
  <updates>
    <add>
      <returnSentItemsId>1</returnSentItemsId>
      <options/>
    </add>
    <update>
      <returnSentItemsId>1</returnSentItemsId>
      <subject>Posted Item - changed</subject>
      <message length="44">VGhpcyBpcyBhIHBvc3RlZCBpdGVtLgoKCmNo
        YW5nZWQA</message>
    </update>
  </updates>
</modifyItemRequest>

<modifyItemResponse>
  <modified>2012-09-14T21:13:29Z</modified>
  <status>
    <code>0</code>
  </status>
</modifyItemResponse>
```

## Modify a Contact

```
<modifyItemRequest>
  <id>406BD9A1.AutoDomain.AutoPO1.104.1363230.1.E3.1@56:
    3B38B6FA.AutoDomain.AutoPO1.104.1363230.1.1.1@53</id>
  <updates>
    <add>
      <phone>1-1111</phone>
      <emailList primary="jdoe@novell.com">
        <email>jdoe@novell.com</email>
      </emailList>
      <phoneList default="2-2222">
        <phone type="Office">1-1111</phone>
        <phone type="Home">2-2222</phone>
        <phone type="Mobile">3-3333</phone>
        <phone type="Fax">4-4444</phone>
        <phone type="Pager">5-5555</phone>
      </phoneList>
      <email>jdoe@novell.com</email>
    </add>
    <delete>
      <phone>1-1111</phone>
```

```
<emailList primary="jdoe@novell.com"
  <email>jdoe@novell.com</email>
</emailList>
<phoneList>
  <phone type="Office">1-1111</phone>
  <phone type="Home">2-2222</phone>
  <phone type="Mobile">3-3333</phone>
  <phone type="Fax">4-4444</phone>
  <phone type="Pager">5-5555</phone>
</phoneList>
  <email>jdoe@novell.com</email>
</delete>
</updates>
</modifyItemRequest>
```

## modifyItemsRequest

Modifies the selected items with the given updates. The modifyItemsRequest is similar to [modifyItemRequest](#). The only difference is that it can process many items in one request.

# modifyJunkEntryRequest

Modifies an existing junk mail entry.

## Request

```
<modifyJunkEntryRequest>
  <entry type="types:JunkEntry"/>
</modifyJunkEntryRequest>
```

## Response

```
<modifyJunkEntryResponse>
  <status type="types:Status"/>
</modifyJunkEntryResponse>
```

## Elements

### entry

(required) Specifies the changes to the existing item.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<modifyJunkEntryRequest>
  <entry>
    <id>44FE8B4B.domain1.pol.104.16E3837.1.106.1@66</id>
    <match>user2222@block.net</match>
    <matchType>domain</matchType>
    <listType>trust</listType>
    <version>0</version>
  </entry>
</modifyJunkEntryRequest>

<modifyJunkEntryResponse>
  <status>
    <code>0</code>
  </status>
</modifyJunkEntryResponse>
```

# modifyJunkMailSettingsRequest

Modifies the junk mail settings.

## Request

```
<modifyJunkMailSettingsRequest>
  <settings type="types:SettingsList"/>
</modifyJunkMailSettingsRequest>
```

## Response

```
<modifyJunkMailSettingsResponse>
  <status type="types:Status"/>
</modifyJunkMailSettingsResponse>
```

## Elements

### settings

Specifies the junk mail setting to be modified.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<modifyJunkMailSettingsRequest>
  <settings>
    <setting>
      <field>useJunkList</field>
      <value>1</value>
    </setting>
    <setting>
      <field>usePAB</field>
      <value>1</value>
    </setting>
    <setting>
      <field>persistence</field>
      <value>7</value>
    </setting>
  </settings>
</modifyJunkMailSettingsRequest>

<modifyJunkMailSettingsResponse>
  <status>
    <code>0</code>
  </status>
</modifyJunkMailSettingsResponse>
```

# modifyPasswordRequest

Modifies a user's password.

## Request

```
<modifyPasswordRequest>
  <old type="string"/>
  <new type="string"/>
</modifyPasswordRequest>
```

## Response

```
<modifyPasswordResponse>
  <status type="types:Status"/>
</modifyPasswordResponse>
```

## Elements

### old

Specifies the text of the old password.

### new

Specifies the text of the new password.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<modifyPasswordRequest>
  <old>ul</old>
  <new>new</new>
</modifyPasswordRequest>

<modifyPasswordResponse>
  <status>
    <code>0</code>
  </status>
</modifyPasswordResponse>
```

# modifyNotifyRequest

Modifies a NotifyEntry element. See [createNotifyRequest](#).

## Request

```
<modifyNotifyRequest>
  <id type="types:uid" />
  <updates type="types:NotifyEntryChanges" />
</modifyNotifyRequest>
```

## Response

```
<modifyNotifyResponse>
  <status type="types:Status" />
</modifyNotifyResponse>
```

## Elements

### id

Specifies the ID of the NotifyEntry element to change.

### code

Specifies the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<modifyNotifyRequest>
  <id>4D3D6590.domain.P01.100.1776172.1.25C8C.1@78</id>
  <updates>
    <update type="NotifyEntry">
      <displayName>Jay E Parker</displayName>
      <rights>
        <alarm>false</alarm>
        <notification>true</notification>
      </rights>
    </update>
  </updates>
</modifyNotifyRequest>

<modifyNotifyResponse>
  <status>
    <code>0</code>
  </status>
</modifyNotifyResponse>
```

# modifyProxyAccessRequest

Modifies the rights a user has to the mailbox.

## Request

```
<modifyProxyAccessRequest>
  <id type="types:uid"/>
  <updates type="types:AccessRightChanges"/>
</modifyProxyAccessRequest>
```

## Response

```
<modifyProxyAccessResponse>
  <status type="types:Status"/>
</modifyProxyAccessResponse>
```

### id

Specifies the ID of the user for which you want to change proxy rights.

### updates

Specifies the rights to change.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<modifyProxyAccessRequest>
  <id>42F9A600175B0000890F6E00D5004E00@60</id>
  <updates>
    <add>
      <appointment>
        <write>1</write>
      </appointment>
      <mail>
        <write>1</write>
      </mail>
      <misc>
        <alarms>1</alarms>
        <notify>1</notify>
        <readHidden>1</readHidden>
        <setup>1</setup>
      </misc>
    </add>
  </updates>
</modifyProxyAccessRequest>
```



```
        <write>1</write>
    </note>
    <task>
        <write>1</write>
    </task>
</add>
</updates>
</modifyProxyAccessRequest>

<modifyProxyAccessResponse>
    <status>
        <code>0</code>
    </status>
</modifyProxyAccessResponse>
```

# modifySettingsRequest

Modifies a user's personal settings. Provide the field and value for each setting that you want to change. If an administrator has locked a client option setting, the setting has a locked element that cannot be changed.

## Request

```
<modifySettingsRequest>
  <settings type="types:SettingsList" />
</modifySettingsRequest>
```

## Response

```
<modifySettingsResponse>
  <status type="types:Status" />
</modifySettingsResponse>
```

## Elements

### settings

Specifies the setting name and new value.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<modifySettingsRequest>
  <settings>
    <setting>
      <field>mailPriority</field>
      <value>Low</value>
    </setting>
  </settings>
</modifySettingsRequest>

<modifySettingsResponse>
  <status>
    <code>0</code>
  </status>
</modifySettingsResponse>
```

# modifySignaturesRequest

Modifies the specified user's signature.

## Request

```
<modifySignaturesRequest>  
  <updates type="types:Signatures"/>  
</modifySignaturesRequest>
```

## Response

```
<modifySignaturesResponse>  
  <status type="types:Status"/>  
</modifySignatureResponse>
```

## Elements

### updates

Specifies the signatures to modify.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

# moveItemRequest

Copies (links) or moves an item from one container into another container. All mail item types can be copied or moved from one folder to another. All address book item types can be copied or moved from one personal address book to another. In GroupWise, an item can appear in more than one container. If an item appears in more than one container, it is called a linked item.

## Request

```
<moveItemRequest>
  <id type="types:uid"/>
  <container type="types:uid"/>
  <from type="types:uid"/>
  <before type="types:uid"/>
  <after type="types:uid"/>
  <under type="types:uid"/>
  <recurrenceAllInstances type="unsignedInt"/>
</moveItemRequest>
```

## Response

```
<moveItemResponse>
  <status type="types:Status"/>
  <id type="types:uid"/>
</moveItemResponse>
```

## Elements

### id

Specifies the uid of the item that you want to move or copy.

### container

Specifies the destination container where the moved or copied item appears.

### from

Specifies the from container uid to move an item from one container to another container. If you want to copy an item in one container to another container, do not provide the from uid.

### before

Specifies the location of the item before it is moved. This element is used to order items in the checklist folder. If you supply item IDs for the before and after elements, the item is positioned between the two items. If you do not supply values for the before and after elements, the item is placed last in the checklist folder. You can specify “first” to place the item first in the list or “last” to place the item last in the list.

**after**

Specifies the location of the item after it is moved. This element is used to order items in the checklist folder. If you supply item IDs for the before and after elements, the item is positioned between the two items. If you do not supply values for the before and after elements, the item is placed last in the checklist folder. You can specify “first” to place the item first in the list or “last” to place the item last in the list.

**code**

Returns the error number related to the event. 0 indicates that the request was successful.

**status**

Returns the success or failure of the method.

**under**

The under element controls making an item a subtask item. The item referenced by id will be put under the item identified by under.

**recurrenceAllInstances**

Passes the recurrenceKey value to move all instances.

**Id (in response)**

After the item is moved, the new item id is returned.

## Example

The following example show different uses of moveItemRequest.

### Move

```
<moveItemRequest>
  <id>44EC25B2.domain1.pol.100.16E3837.1.F75.1@1:44EAEB8F.
    domain1.pol.100.16E3837.1.F6F.1@35</id>
  <container>C.domain1.pol.100.0.1.0.1@21</container>
  <from>44EAEB8F.domain1.pol.100.16E3837.1.F6F.1@35</from>
</moveItemRequest>
```

### Copy (Link)

```
<moveItemRequest>
  <id>44EC25B2.domain1.pol.100.16E3837.1.F75.1@1:44EAEB8F.
    domain1.pol.100.16E3837.1.F6F.1@35</id>
  <container>450A68B9.domain1.pol.100.16E3837.1.FCC.1@13</container>
</moveItemRequest>
```

# moveItemsRequest

Copies (links) multiple items for a source (from) container to a destination (container). --**For GroupWise 2012 and later.**

## Request

```
<moveItemsRequest>
  <item type="types:MoveItem"/>
</moveItemsRequest>
```

## Response

```
<moveItemResponse>
  <id type="types:uid"/>
  <status type="types:Status"/>
</moveItemResponse>
```

## Elements

### item

MoveItem element. (Contains id, container and from elements.)

### id

On input, specifies the uid of the item that you want to move or copy.

On output, specifies the id of the item in the new container.

### container

Specifies the destination container where the moved or copied item appears.

### from

Specifies the from container uid to move an item from one container to another container. If you want to copy an item in one container to another container, do not provide the from uid.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<moveItemsRequest>
  <item>
    <id>4D3EE0F2.domain.P01.100.1776172.1.25C9F.1@1:
      C.domain.P01.100.0.1.0.1@21</id>
    <container>7.domain.P01.100.0.1.0.1@16</container>
    <from>C.domain.P01.100.0.1.0.1@21</from>
  </item>
  <item>
    <id>4D3EE101.domain.P01.100.1776172.1.25CA2.1@1:
      C.domain.P01.100.0.1.0.1@21</id>
    <container>7.domain.P01.100.0.1.0.1@16</container>
    <from>C.domain.P01.100.0.1.0.1@21</from>
  </item>
</moveItemsRequest>

<moveItemsResponse>
  <id>4D3EE0F2.domain.P01.100.1776172.1.25C9F.1@1:
    7.domain.P01.100.0.1.0.1@16</id>
  <id>4D3EE101.domain.P01.100.1776172.1.25CA2.1@1:
    7.domain.P01.100.0.1.0.1@16</id>
  <status>
    <code>0</code>
  </status>
</moveItemsResponse>
```

# positionCursorRequest

Positions the cursor for the next read. The positionCursorRequest can be called only if a cursor has been created by calling [createCursorRequest](#).

## Request

```
<positionCursorRequest>
  <container type="types:uid"/>
  <cursor type="int"/>
  <seek type="types:CursorSeek"/>
  <offset type="int"/>
</positionCursorRequest>
```

## Response

```
<positionCursorResponse>
  <status type="types:Status"/>
</positionCursorResponse>
```

## Elements

### container

(required) Specifies the ID of the container.

### cursor

Specifies the value returned from createCursorRequest.

### seek

Specifies the anchor position of the cursor: current, start, or end.

### offset

Specifies the offset from the anchor position of the cursor.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.



## Example

```
<positionCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1184919325</cursor>
  <seek>end</seek>
  <offset>5</offset>
</positionCursorRequest>

<positionCursorResponse>
  <status>
    <code>0</code>
  </status>
</positionCursorResponse>
```

# purgeDeletedItemsRequest

Purges (permanently deletes) all items in the Trash folder.

## Request

```
<purgeDeletedItemsRequest />
```

## Response

```
<purgeDeletedItemsResponse>  
  <status type="types:Status" />  
</purgeDeletedItemsResponse>
```

## Elements

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<purgeDeletedItemsRequest />  
  
<purgeDeletedItemsResponse>  
  <status>  
    <code>0</code>  
  </status>  
</purgeDeletedItemsResponse>
```

# purgeRequest

Purges (permanently deletes) the selected item from the container.

## Request

```
<purgeRequest>
  <items type="types:ItemRefList">
    <recurrenceAllInstances type="unsignedInt"/>
  </purgeRequest>
```

## Response

```
<purgeResponse>
  <status type="types:Status"/>
</purgeResponse>
```

## Elements

### items

Specifies the items to be purged from the user's account.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

### recurrenceAllInstances

Specifies that you want the purge to be applied to all recurring items if you pass the recurrenceKey.

## Example

```
<purgeRequest>
  <items>
    <item>450FE01C.domain1.pol.100.16E3837.1.102A.1@1:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
</purgeRequest>

<purgeResponse>
  <status>
    <code>0</code>
  </status>
</purgeResponse>
```

# readCursorRequest

Reads from the container at the current cursor position. For more information please see [Getting Items](#).

## Request

```
<readCursorRequest>
  <container type="types:uid"/>
  <cursor type="int"/>
  <forward type="boolean"/>
  <position type="types:CursorSeek"/>
  <count type="int"/>
</readCursorRequest>
```

## Response

```
<readCursorResponse>
  <items type="types:ItemList"/>
  <status type="types:Status"/>
</readCursorResponse>
```

## Elements

### container

Specifies the ID of the container.

### cursor

Specifies the value returned from [createCursorRequest](#).

### forward

Specifies whether to read forward or backward.

### position

Specifies the anchor position of the cursor: current, start, or end.

### count

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

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<readCursorRequest>
  <container>7.domain1.pol.100.0.1.0.1@16</container>
  <cursor>1184907119</cursor>
  <forward>false</forward>
  <position>current</position>
  <count>5</count>
</readCursorRequest>

<readCursorResponse>
  <items>
    <item>
      <id>450AD3EB.domain1.pol.100.16E3837.1.101E.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:34Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>20</subject>
    </item>
    <item>
      <id>450AD3E5.domain1.pol.100.16E3837.1.101C.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:33Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>19</subject>
    </item>
    <item>
      <id>450AD3E0.domain1.pol.100.16E3837.1.101A.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:33Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>18</subject>
    </item>
    <item>
      <id>450AD3DA.domain1.pol.100.16E3837.1.1018.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:32Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>17</subject>
    </item>
    <item>
      <id>450AD3D4.domain1.pol.100.16E3837.1.1016.1@1:7.
        domain1.pol.100.0.1.0.1@16</id>
      <modified>2012-09-19T19:26:32Z</modified>
      <container>7.domain1.pol.100.0.1.0.1@16</container>
      <source>received</source>
      <subject>16</subject>
    </item>
  </items>
</readCursorResponse>
```

# removeCustomDefinitionRequest

Removes custom field definitions. It does not remove field and values within an item. It can take a long time to execute.

## Request

```
<removeCustomDefinitionRequest>  
  <customs type="types:CustomList" />  
  <books type="boolean" />  
  <doAsynchronous type="boolean" />  
</removeCustomDefinitionRequest>
```

## Response

```
<removeCustomDefinitionResponse>  
  <status type="types:Status" />  
</removeCustomDefinitionResponse>
```

## Elements

### customs

Specifies the name of the custom field definition to remove.

### books

Specifies whether to delete custom definitions from personal address book entries (pass 1). Otherwise, the definition is deleted from mail items.

### doAsynchronous

Specifies whether the doAsynchronous element removes the definitions in the background (True).

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeCustomDefinitionRequest>
  <customs>
    <custom>
      <field>custom3</field>
    </custom>
  </customs>
  <books>false</books>
  <doAsynchronous>false</doAsynchronous>
</removeCustomDefinitionRequest>

<removeCustomDefinitionResponse>
  <status>
    <code>0</code>
  </status>
</removeCustomDefinitionResponse>
```

# removeItemRequest

Removes the specified item. Mailbox items (mail, appointments, notes, task, and phone) are moved to the Trash folder.

## Request

```
<removeItemRequest>
  <container type="types:uid"/>
  <id type="types:uid"/>
</removeItemRequest>
```

## Response

```
<removeItemResponse>
  <status type="types:Status"/>
</removeItemResponse>
```

## Elements

### container

Specifies the container. If the container is not specified, the item is removed from all of the containers and stored in the Trash. If the container is specified, the item is removed only from that specific container and the link is stored in the Trash. This functionality does not apply to address book items.

### id

Specifies the ID of the item to move to the Trash folder.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeItemRequest>
  <container>A.dom1.pol.100.0.1.0.1@19</container>
  <id>4062AED9.dom1.pol.100.164677A.1.1AE.1@4:A.
    dom1.pol.100.0.1.0.1@19</id>
</removeItemRequest>
```



# removeItemsRequest

Removes the specified items. Mailbox items (mail, appointments, notes, task, and phone) are moved to the Trash folder. This method is similar to [removeItemRequest](#), except that it works on many items.

## Request

```
<removeItemsRequest>
  <container type="types:uid"/>
  <items type="types:ItemRefList"/>
  <recurrenceAllInstances type="unsignedInt"/>
</removeItemsRequest>
```

## Response

```
<removeItemsResponse>
  <status type="types:Status"/>
</removeItemsResponse>
```

## Elements

### container

Specifies the container. If the container is not specified, the item is removed from all of the containers and stored in the Trash. If the container is specified, the item is removed only from that specific container and the link is stored in the Trash. Does not apply to address book items.

### items

Specifies a list of items to move to the trash folder.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

### recurrenceAllInstances

Specifies that you want the purge to be applied to all recurring items if you pass the recurrenceKey.

## Example

```
<removeItemsRequest>
  <container>450A68B9.domain1.pol.100.16E3837.1.FCC.1@13</container>
  <items>
    <item>450AB88A.domain1.pol.100.16E3837.1.FF6.1@2:450A68B9.
      domain1.pol.100.16E3837.1.FCC.1@13</item>
    <item>450AB88A.domain1.pol.100.16E3837.1.FF6.1@2:450A68B9.
      domain1.pol.100.16E3837.1.FCC.1@13</item>
  </items>
</removeItemsRequest>

<removeItemsResponse>
  <status>
    <code>0</code>
  </status>
</removeItemsResponse>
```

# removeJunkEntryRequest

Removes an address or a domain from the junk, block, or trust lists.

## Request

```
<removeJunkEntryRequest>
  <id type="types:uid"/>
</removeJunkEntryRequest>
```

## Response

```
<removeJunkEntryResponse>
  <status type="types:Status"/>
</removeJunkEntryResponse>
```

## Elements

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeJunkEntryRequest>
  <id>44FE8B4B.domain1.pol.104.16E3837.1.106.1@66</id>
</removeJunkEntryRequest>

<removeJunkEntryResponse>
  <status>
    <code>0</code>
  </status>
</removeJunkEntryResponse>
```

# removeMembersRequest

Removes the specifies members from a group. Items are removed only from the group itself, not deleted from the address book.

## Request

```
<removeMembersRequest>
  <container type="types:uid"/>
  <members type="types:GroupMemberList"/>
</removeMembersRequest>
```

## Response

```
<removeMembersResponse>
  <status type="types:Status"/>
</removeMembersResponse>
```

## Elements

### container

Specifies the group ID.

### members

Specifies the member to remove from the group.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeMembersRequest>
  <container>42C510EA.domain1.pol.104.16E3837.1.3.1@53</container>
  <members>
    <member>
      <id>44F855B6.domain1.pol.104.16E3837.1.FB.1@56:42C510EA.
        domain1.pol.104.16E3837.1.3.1@53</id>
      <email>dd@novell.com</email>
    </member>
  </members>
</removeMembersRequest>

<removeMembersResponse>
  <status>
    <code>0</code>
  </status>
</removeMembersResponse>
```

# removeProxyAccessRequest

Removes a person from the list of users that can proxy to the mailbox.

## Request

```
<removeProxyAccessRequest>
  <id type="types:uid"/>
</removeProxyAccessRequest>
```

## Response

```
<removeProxyAccessResponse>
  <status type="types:Status"/>
</removeProxyAccessResponse>
```

## Elements

### id

Specifies the ID of the user to remove from the proxy access list.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeProxyAccessRequest>
  <id>66301550-175B-0000-890F-6E00D5004E00@60</id>
</removeProxyAccessRequest>

<removeProxyAccessResponse>
  <status>
    <code>0</code>
  </status>
</removeProxyAccessResponse>
```

# removeNotifyRequest

Remove a NotifyEntry element. See [createNotifyRequest](#). --For GroupWise 8.0 SP1 and later.

## Request

```
<removeNotifyRequest>
  <id type="types:uid"/>
</removeNotifyRequest>
```

## Response

```
<removeNotifyResponse>
  <status type="types:Status" />
</removeNotifyResponse>
```

## Elements

### id

Specifies the ID of the NotifyEntry element to remove.

### code

Specifies the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeNotifyRequest>
  <id>4D3D6590.domain.P01.100.1776172.1.25C8C.1@78</id>
</removeNotifyRequest>

<removeNotifyResponse>
  <status>
    <code>0</code>
  </status>
</removeNotifyResponse>
```

# removeSignatureRequest

Removes the specified signatures.

## Request

```
<removeSignatureRequest>
  <id type="types:uid"/>
  <all type="boolean"/>
  <global type="boolean"/>
</removeSignatureRequest>
```

## Response

```
<removeSignatureResponse>
  <status type="types:Status"/>
</removeSignatureResponse>
```

## Elements

### id

Specifies the ID of the signature to be deleted.

### all

Specifies whether to remove all the signatures (True).

### global

Is not used.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<removeSignatureRequest>
  <id>44FE9BAA.domain1.pol.100.16E3837.1.F95.1@63</id>
  <all>false</all>
  <global>false</global>
</removeSignatureRequest>

<removeSignatureResponse>
  <status>
    <code>0</code>
  </status>
</removeSignatureResponse>
```

# replyRequest

Replies to an existing message.

## Request

```
<replyRequest>
  <id type="types:uid"/>
  <view type="string"/>
</replyRequest>
```

## Response

```
<replyResponse>
  <item type="types:Item"/>
  <status type="types:Status"/>
</replyResponse>
```

## Elements

### id

Specifies the ID of the item you want to reply to.

### view

Specifies whether you want the original message body returned in the reply call:

- ♦ message
- ♦ message/RTF
- ♦ recipients-Replies to all recipients. Otherwise, the reply applies to the sender only.

### item

Returns an item that can be modified and passed to sendRequest to complete the reply.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Remarks

Replying to a message is a two-step process. The first step is to call the reply method with the ID of the item that you want to reply to. A new mail item is then returned with the original subject and other fields that are specified in the view. An element called linkInfo is returned in this item.

The second step is to create a new item to send. The linkInfo element needs to be passed unchanged in the sendRequest. The new item can then be modified at will. For example, you might want to prepend "Re:" to the beginning of the subject. The message body can also be modified and attachments added.



## Example

```
<replyRequest>
  <id>450FE559.domain1.pol.100.16E3837.1.1030.1@1:7.
    domain1.pol.100.0.1.0.1@16</id>
  <view/>
</replyRequest>

<replyResponse>
  <item type="Mail">
    <source>sent</source>
    <subject>Re: Reply - threaded message</subject>
    <distribution/>
    <link>
      <id>0:4144</id>
      <type>reply</type>
    </link>
  </item>
  <status>
    <code>0</code>
  </status>
</replyResponse>
```

# resendRequest

Resends an item. This is a two step process like the forwardRequest. You call resendRequest. You get back a Mail item with a LinkInfo element. You pass that element unchanged in the new item. If you want to get back the original message body, you pass “message” or “message/RTF” in the view element of the resendRequest. --For GroupWise 2012 and later.

## Request

```
<resendRequest>
  <id type="types:uid"/>
  <view type="string"/>
</delegateRequest>
```

## Response

```
<resendResponse>
  <item type="types:Item"/>
  <status type="types:Status"/>
</delegateResponse>
```

## Elements

### id

Specifies the ID of the item to be resent.

### view

Controls what is returned in the Mail item.

### item

Mail item with the information for the resend.

### code

Returns the error number related to the request. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example of Original Note

```
<sendItemRequest>
  <item type="Note">
    <returnSentItemsId>true</returnSentItemsId>
    <subject>Note to resend</subject>
    <distribution>
      <to>Preston Stepehnson; Jay Parker</to>
      <recipients>
        <recipient type="Recipient">
          <displayName>Preston Stephenson</displayName>
          <email>pstehenson@prestons.novell.com</email>
          <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
        </recipient>
        <recipient type="Recipient">
          <displayName>Jay Parker</displayName>
          <email>jparker@prestons.novell.com</email>
          <uuid>C974CA00-0B55-0000-B976-8E90E5B002BD</uuid>
        </recipient>
      </recipients>
    </distribution>
    <message>
      <part length="0"
        offset="0">VGhpcyBpcyB0aGUgb3JpZ25hbCBtZXNzYWdlIGJvZHku</part>
    </message>
    <startDate>2011-01-29</startDate>
  </item>
</sendItemRequest>

<sendItemResponse>
  <id>4D417E7B.domain.P01.100.1776172.1.25D1B.1@2:
    7.domain.P01.100.0.1.0.1@16</id>
  <id>4D417E7B.domain.P01.100.1776172.1.25D1C.1@2:
    7.domain.P01.100.0.1.0.1@16</id>
  <status>
    <code>0</code>
  </status>
</sendItemResponse>
```

## Example of Get Note

```
<getItemRequest>
  <id>4D417E7B.domain.P01.100.1776172.1.25D1B.1@2:
    7.domain.P01.100.0.1.0.1@16</id>
  <view>default message recipients</view>
</getItemRequest>

<getItemResponse>
  <item type="Note">
    <id>4D417E7B.domain.P01.100.1776172.1.25D1B.1@2:
      7.domain.P01.100.0.1.0.1@16</id>
    <sid>154907</sid>
    <version>3</version>
    <modified>2011-01-27T21:17:31Z</modified>
    <container sid="5">A.domain.P01.100.0.1.0.1@19</container>
    <created>2011-01-27T21:17:31Z</created>
    <status>
      <accepted>1</accepted>
      <opened>1</opened>
      <read>1</read>
    </status>
    <msgId>4D417E7B.domain.P01.200.20000C2.1.42820.1</msgId>
    <clientMessageId>4D41E0EB.141 : 194 : 18298</clientMessageId>
    <source>sent</source>
    <delivered>2011-01-27T21:17:31Z</delivered>
    <security>Normal</security>
    <subject>Note to resend</subject>
    <distribution>
      <from>
        <displayName>Preston Stephenson</displayName>
        <email>pstephenson@prestons.provo.novell.com</email>
        <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
      </from>
      <to>Preston Stepehnson; Jay Parker</to>
      <recipients>
        <recipient>
          <displayName>Jay Parker</displayName>
          <email>jparker@prestons.provo.novell.com</email>
          <uuid>C974CA00-0B55-0000-B976-8E90E5B002BD</uuid>
          <distType>TO</distType>
          <recipType>User</recipType>
          <domain>domain</domain>
          <postOffice>P01</postOffice>
          <idomain>prestons.provo.novell.com</idomain>
        </recipient>
        <recipient>
          <displayName>Preston Stephenson</displayName>
          <email>pstephenson@prestons.provo.novell.com</email>
          <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
          <distType>TO</distType>
          <recipType>User</recipType>
          <domain>domain</domain>
          <postOffice>P01</postOffice>
          <idomain>prestons.provo.novell.com</idomain>
        </recipient>
      </recipients>
    </distribution>
  </item>
</getItemResponse>
```

```

        </recipient>
    </recipients>
    <sendoptions>
        <statusTracking>None</statusTracking>
    </sendoptions>
</distribution>
<message>
    <part contentType="text/plain"
        length="44">VGhpcyBpcyB0aGUgb3JpZ25hbCBtZXNzYWdlIGJvZHku</part>
</message>
<options>
    <priority>Standard</priority>
</options>
<size>809</size>
<retentionModified>2011-01-27T21:17:31Z</retentionModified>
<iCalId>2011-01-27T21:17:31Z_477A00C2@prestons.provo.novell.com</
iCalId>
    <startDate>2011-01-29</startDate>
    <timezone>
        <daylight>
            <month>3</month>
            <dayOfWeek occurrence="Second">Sunday</dayOfWeek>
            <hour>0</hour>
            <minute>0</minute>
            <offset>-21600</offset>
        </daylight>
        <standard>
            <month>11</month>
            <dayOfWeek occurrence="First">Sunday</dayOfWeek>
            <hour>0</hour>
            <minute>0</minute>
            <offset>-25200</offset>
        </standard>
    </timezone>
</item>
<status>
    <code>0</code>
</status>
</getItemResponse>

```

## Example of Resend Request

```
<resendRequest>
  <id>4D417E7B.domain.P01.100.1776172.1.25D1B.1@2:
    7.domain.P01.100.0.1.0.1@16</id>
</resendRequest>

<resendResponse>
  <item type="Mail">
    <sid>154907</sid>
    <source>sent</source>
    <delivered>2011-01-27T21:17:31Z</delivered>
    <subject>Note to resend</subject>
    <distribution>
      <from>
        <displayName>Preston Stephenson</displayName>
      </from>
    </distribution>
    <link>
      <id>4D417E7B.domain.P01.100.1776172.1.25D1B.1</id>
      <type>resend</type>
    </link>
  </item>
  <status>
    <code>0</code>
  </status>
</resendResponse>
```

## Example of New Send

```
<sendItemRequest>
  <item type="Note">
    <subject>subject changed for resend</subject>
    <distribution>
      <from type="From">
        <displayName>Preston Stephenson</displayName>
        <email>pstephenson@prestons.provo.novell.com</email>
        <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
      </from>
      <to>Preston Stepehnson; Jay Parker</to>
      <recipients>
        <recipient type="Recipient">
          <displayName>Jay Parker</displayName>
          <email>jparker@prestons.provo.novell.com</email>
          <uuid>C974CA00-0B55-0000-B976-8E90E5B002BD</uuid>
        </recipient>
        <recipient type="Recipient">
          <displayName>Preston Stephenson</displayName>
          <email>pstephenson@prestons.provo.novell.com</email>
          <uuid>2D16B180-0922-0000-846C-82BD300F9E1C</uuid>
        </recipient>
      </recipients>
    </distribution>
  </item>
  <message>
```

```

    <part length="0"
      offset="0">VGhpcyBpcyB0aGUgb3JpZ25hbCBtZXNzYWdlIGJvZH
      kuClRoaxMgaXMgdGh1IHNLy29uZCBsaW5lLg==</part>
  </message>
  <link>
    <id>4D417E7B.domain.PO1.100.1776172.1.25D1B.1</id>
    <type>resend</type>
  </link>
  <startDate>2011-01-29</startDate>
  <timezone>
    <daylight>
      <month>3</month>
      <day>0</day>
      <dayOfWeek occurrence="Second">Sunday</dayOfWeek>
      <hour>0</hour>
      <minute>0</minute>
      <offset>-21600</offset>
    </daylight>
    <standard>
      <month>11</month>
      <day>0</day>
      <dayOfWeek occurrence="First">Sunday</dayOfWeek>
      <hour>0</hour>
      <minute>0</minute>
      <offset>-25200</offset>
    </standard>
  </timezone>
</item>
</sendItemRequest>

<sendItemResponse>
  <id>4D417E7C.domain.PO1.100.1776172.1.25D1E.1@2:
    7.domain.PO1.100.0.1.0.1@16</id>
  <status>
    <code>0</code>
  </status>
</sendItemResponse>

```

# resolveRequest

Resolves or validates recipients in the GroupWise system. In other words, it confirms the existence of recipients in the GroupWise system.

## Request

```
<resolveRequest>
  <recipients type="types:RecipientList"/>
</resolveRequest>
```

## Response

```
<resolveResponse>
  <recipients type="types:RecipientList"/>
  <status type="types:Status"/>
</resolveResponse>
```

## Elements

### recipients (inbound)

Recipients to validate.

### recipients (outbound)

List of valid or invalid recipients.

### status

Returns the success or failure of the method.



# restoreItemRequest

The restoreItemRequest is used to restore an item that has been archived by a third party using the archive WSDL and schema. See the [stubbing/archive](#) documentation for more information.

## Request

```
<restoreItemRequest>
  <id type="types:uid"/>
  <item type="types:Items"/>
</restoreItemRequest>
```

## Response

```
<restoreItemResponse>
  <id type="types:uid"/>
  <status type="types:Status"/>
</restoreItemResponse>
```

## Elements

### id

Stubbed item id.

### item

Item to be restored. It is the item that was archived by the third party.

### status

Returns the success or failure of the method.

## Remarks

GroupWise has a phantom item called a stub that represent the archived item. When an item is restored with restoreItemRequest, the stub needs to be removed from the mailbox and replaced by the item.

# retractRequest

Retracts calendar items.

## Request

```
<retractRequest>
  <items type="types:ItemRefList" />
  <comment type="string" />
  <retractingAllInstances type="boolean" />
  <retractCausedByResend type="boolean" />
  <retractType type="types:RetractType" />
  <recurrenceAllInstances type="unsignedInt" />
</retractRequest>
```

## Response

```
<retractResponse>
  <status type="types:Status" />
</retractResponse>
```

## Elements

### items

Specifies the calendar items to retract.

### comment

Specifies that the comment is included in the notice.

### retractCausedByResend

Specifies that a notice is not sent because it is suppressed and replaced with the new calendar item (True).

### retractingAllInstances

Specifies that a single notice is placed in the recipient's in box (True), rather than a notice for each instance.

### retractType

Specifies where to retract the item:

myMailbox-Retract only the items in the sender's mailbox.

recipientMailboxes-Retract only the items in the recipient's mailbox.

allMailboxes-Retract the items in both the sender's and recipient's mailboxes.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## **recurrenceAllInstances**

Specifies that you want the purge to be applied to all recurring items if you pass the recurrenceKey.

## **Remarks**

If a recipient has opened or accepted a calendar item, a notice or mail item is placed in the recipient's In box when the item is retracted.

## **Example**

```
<retractRequest>
  <items>
    <item>4510FFC8.domain1.pol.100.16E3837.1.103D.1@4:7.
      domain1.pol.100.0.1.0.1@16</item>
  </items>
  <comment>comment</comment>
  <retractingAllInstances>true</retractingAllInstances>
  <retractCausedByResend>true</retractCausedByResend>
  <retractType>allMailboxes</retractType>
</retractRequest>

<retractResponse>
  <status>
    <code>0</code>
  </status>
</retractResponse>
```

# sendItemRequest

Sends a distributed item. It is also used to create draft and personal items.

## Request

```
<sendItemRequest>
  <item type="types:Item"/>
</sendItemRequest>
```

## Response

```
<sendItemResponse>
  <id type="types:uid"/>
  <status type="types:Status"/>
</sendItemResponse>
```

## Elements

### item

Specifies all the parameters for the send. Valid item types that can be sent are: Mail, Appointments, Notes, Tasks, and Phone items.

### id

Specifies the ID of the item that was created in the sent item folder. For example, if you send an appointment and you are one of the recipients, the ID is the ID of the sent item and not the received item. If the item is a recurring item, sendItemResponse returns an ID for each recurring instance created in GroupWise.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<sendItemRequest>
  <item type="Appointment">
    <source>sent</source>
    <class>Public</class>
    <acceptLevel>Busy</acceptLevel>
    <subject>Sending Example</subject>
    <distribution>
      <recipients>
        <recipient>
          <displayName>u2</displayName>
          <email>u2@phantom.com</email>
          <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
          <distType>TO</distType>
        </recipient>
      </recipients>
    </distribution>
  </item>
</sendItemRequest>
```

```

        </recipients>
        <sendoptions>
            <requestReply/>
            <statusTracking>All</statusTracking>
            <updateFrequentContacts>0</updateFrequentContacts>
        </sendoptions>
    </distribution>
    <message>
        <part length="36">VGhpcyBpcyB0aGUgbWVzc2FnZSBib2R5LgA=</part>
    </message>
    <options>
        <priority>Standard</priority>
    </options>
    <startDate>20120920T160000Z</startDate>
    <endDate>20120920T170000Z</endDate>
    <alarm types:enabled="1">5</alarm>
    <place>place</place>
</types:item>
</sendItemRequest>

<sendItemResponse>
    <status>
        <code>0</code>
    </status>
</sendItemResponse>

```

# setTimestampRequest

Sets the last backup date and last retention date on a mailbox. It can be called only when authenticated by using a trusted application.

## Request

```
<setTimestampRequest>
  <backup type="dateTime"/>
  <retention type="dateTime"/>
  <retentionModified type="boolean"/>
</setTimestampRequest>
```

## Response

```
<setTimestampResponse>
  <status type="types:Status"/>
  <retentionModified type="dateTime"/>
</setTimestampResponse>
```

## Elements

### backup

Specifies the last backup date on a mailbox.

### retention

Specifies the last retention date on a mailbox.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

### 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 timestamp. Suppose an item has a retention timestamp. If a user modifies the item by adding a personal attachment or some other significant change, retention software will skip the item because it has already been retained. RetentionModified is now used to catch an item after the first retention and a significant change has occurred to the item.

## Example

```
<setTimestampRequest>
  <backup>20120920T103000Z</backup>
  <retention>20120920T103000Z</retention>
</setTimestampRequest>

<setTimestampResponse>
  <status>
    <code>0</code>
  </status>
</setTimestampResponse>
```

# startFreeBusySessionRequest

Starts a free/busy search on the specified users. On a successful startFreeBusySessionResponse, a freeBusySessionId is returned. The freeBusySessionId is used with the other free/busy methods: [getFreeBusyRequest](#) and [closeFreeBusySessionRequest](#).

## Request

```
<startFreeBusySessionRequest>
  <users type="types:FreeBusyUserList" />
  <startDate type="dateTime" />
  <endDate type="dateTime" />
</startFreeBusySessionRequest>
```

## Response

```
<startFreeBusySessionResponse>
  <freeBusySessionId type="int" />
  <status type="types:Status" />
</startFreeBusySessionResponse>
```

## Elements

### user

Specifies the list of users that you want free/busy information for.

### startDate

Specifies the start of the range of time to search.

### endDate

Specifies the end of the range of time to search.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.



## Example

```
<startFreeBusySessionRequest>
  <users>
    <user>
      <displayName>u2</displayName>
      <email>u2@phantom.com</email>
      <uuid>42F9A600-175B-0000-890F-6E00D5004E00</uuid>
    </user>
  </users>
  <startDate>20120920T060000Z</startDate>
  <endDate>20121011T060000Z</endDate>
</startFreeBusySessionRequest>

<startFreeBusySessionResponse>
  <freeBusySessionId>4163</freeBusySessionId>
  <status>
    <code>0</code>
  </status>
</startFreeBusySessionResponse>
```

# streamedSearchRequest

Used to provide a cursor-like search over a document library. You pass the lastHitId element value from the last streamedSearchResponse in the next streamSearchRequest until you get back an empty list or a null lastHitId element. --For GroupWise 2012 and later.

## Request

```
<streamedSearchRequest>
  <container type="types:uid" />
  <view type="string" />
  <filter type="types:Filter" />
  <count type="int" />
  <lastHitId type="xs:string" />
</streamedSearchRequest>
```

## Response

```
<streamedSearchResponse>
  <items type="types:Items" />
  <lastHitId type="string" />
  <status type="types:Status" />
</streamedSearchResponse>
```

## Elements

### container

(optional) Specifies the library container ID. If the container is not specified, the filter is used. If the filter is not present, an error is returned.

### view

Specifies the elements returned for each item. The view reduces the amount of data returned. If a view is not specified, all item elements are returned.

### filter

Specifies the items to return based on a filter.

### items

Specifies a list of item ids to return.

### count

Specifies the number of items to return. If the count element is -1, all items are returned.

### items

Returns the specified items.

### code

Specifies the error number related to the event. 0 indicates that the request was successful.

## status

Returns the success or failure of the method.

## Example

```
<streamedSearchRequest>
  <container>domain.P01.P01 Library@67</container>
  <filter>
    <element type="FilterEntry">
      <op>contains</op>
      <field>subject</field>
      <value>test</value>
    </element>
  </filter>
  <count>5</count>
</streamedSearchRequest>

<streamedSearchResponse>
  <items>
  ...
</items>
<lastHitId>201326849_121_5</lastHitId>
<status>
  <code>0</code>
</status>
</streamedSearchResponse>

<streamedSearchRequest>
  <container>domain.P01.P01 Library@67</container>
  <count>5</count>
  <lastHitId>201326849_121_5</lastHitId>
</streamedSearchRequest>

<streamedSearchResponse>
  <items>
  ...
</items>
<lastHitId>201327105_15_12</lastHitId>
<status>
  <code>0</code>
</status>
</streamedSearchResponse>

<streamedSearchRequest>
  <container>domain.P01.P01 Library@67</container>
  <count>5</count>
  <lastHitId>201327105_15_12</lastHitId>
</streamedSearchRequest>

<streamedSearchResponse>
  <items>
  ...
</items>
<lastHitId>201327617_22_17</lastHitId>
```

```

    <status>
      <code>0</code>
    </status>
  </streamedSearchResponse>

  <streamedSearchRequest>
    <container>domain.P01.P01 Library@67</container>
    <count>5</count>
    <lastHitId>201327617_22_17</lastHitId>
  </streamedSearchRequest>

  <streamedSearchResponse>
    <items>
      ...
    </items>
    <lastHitId>201327873_20_22</lastHitId>
    <status>
      <code>0</code>
    </status>
  </streamedSearchResponse>

  <streamedSearchRequest>
    <container>domain.P01.P01 Library@67</container>
    <count>5</count>
    <lastHitId>201327873_20_22</lastHitId>
  </streamedSearchRequest>

  <streamedSearchResponse>
    <items>
      ...
    </items>
    <lastHitId>201328385_215_27</lastHitId>
    <status>
      <code>0</code>
    </status>
  </streamedSearchResponse>

  <streamedSearchRequest>
    <container>domain.P01.P01 Library@67</container>
    <count>5</count>
    <lastHitId>201328385_215_27</lastHitId>
  </streamedSearchRequest>

  <streamedSearchResponse>
    <items>
      ...
    </items>
    <lastHitId>201328897_25_32</lastHitId>
    <status>
      <code>0</code>
    </status>
  </streamedSearchResponse>

  <streamedSearchRequest>
    <container>domain.P01.P01 Library@67</container>

```

```

    <count>5</count>
    <lastHitId>201328897_25_32</lastHitId>
</streamedSearchRequest>

<streamedSearchResponse>
  <items>
...
  </items>
  <lastHitId>201328897_9_37</lastHitId>
  <status>
    <code>0</code>
  </status>
</streamedSearchResponse>

<streamedSearchRequest>
  <container>domain.P01.P01 Library@67</container>
  <count>5</count>
  <lastHitId>201328897_9_37</lastHitId>
</streamedSearchRequest>

<streamedSearchResponse>
  <status>
    <code>0</code>
  </status>
</streamedSearchResponse>

```

# stubItemRequest

The stubItemrequest is used to stub/archive an item into a third party database.

## Request

```
<stubItemRequest>
  <id type="types:uid"/>
  <archiveId type="xs:string"/>
</stubItemRequest>
```

## Response

```
<stubItemResponse>
  <id type="types:uid"/>
  <status type="types:Status"/>
</stubItemResponse>
```

## Elements

### id (inbound/request)

Specifies the GroupWise item to be stubbed..

### archiveId

Specifies the third party's archive id..

### id (outbound/response)

Specifies the new stubbed id. This is a new item.

## Remarks

When an item is stubbed via stubItemRequest, GroupWise removes the attachments and message bodies from the item. Saving space is the key benefit of stubbing. In the Windows client, the item will appear with a stubbed icon. To retrieve the item via GroupWise Web Services, the keyword "stubbed" needs to be included in the view.

# unacceptRequest

Marks accepted items as unaccepted or declined.

## Request

```
<unacceptRequest>
  <items type="types:ItemRefList" />
</unacceptRequest>
```

## Response

```
<unacceptResponse>
  <status type="types:Status" />
</unacceptResponse>
```

## Elements

### items

Specifies the items to be marked as unaccepted.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<unacceptRequest>
  <items>
    <item>4510FFB4.domain1.pol.100.16E3837.1.103B.1@4:A.
      domain1.pol.100.0.1.0.1@19</item>
  </items>
  <comment/>
  <acceptLevel/>
</unacceptRequest>

<unacceptResponse>
  <status>
    <code>0</code>
  </status>
</unacceptResponse>
```

# unarchiveRequest

Unarchives item(s) to an archive database.

## Request

```
<archiveRequest>
  <items/>
  <path/>
</archiveRequest>
```

## Response

```
<archiveResponse>
  <path/>
  <status/>
</archiveResponse>
```

## Elements

### items

Specifies the items to unarchive.

### path

On input, specifies the path to use to create the archive directory. If a path is not specified, the POA oftemp directory is used.

On output, returns the path to the archive directory.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

See [archiveRequest](#).



# uncompleteRequest

Marks completed items as uncompleted.

## Request

```
<uncompleteRequest>
  <items type="types:ItemRefList" />
</uncompleteRequest>
```

## Response

```
<uncompleteResponse>
  <status type="types:Status" />
</uncompleteResponse>
```

## Elements

### items

Specifies the items to be marked as not completed.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<uncompleteRequest>
  <items>
    <item>4510FFDF.domain1.pol.100.16E3837.1.103F.1@3:A.
      domain1.pol.100.0.1.0.1@19</item>
  </items>
</uncompleteRequest>

<uncompleteResponse>
  <status>
    <code>0</code>
  </status>
</uncompleteResponse>
```

# updateversionStatusRequest

Updates a document managed by GroupWise Document Management Services.

## Request

```
<updateversionStatusRequest>
  <id type="types:uid"/>
  <event type="types:versionEventType"/>
  <part type="types:SignatureData"/>
</updateversionStatusRequest>
```

## Response

```
<updateversionStatusResponse>
  <part type="types:SignatureData"/>
  <status type="types:Status"/>
</updateversionStatusResponse>
```

## Elements

### id

Specifies the document ID.

### event

Specifies the event type: checkIn, checkOut, resetStatus, and viewed.

### part

Specifies the document itself. The event types that use the part are checkIn, checkOut, and viewed.

### part

Specifies the document itself.

### code

Returns the error number related to the event. 0 indicates that the request was successful.

### status

Returns the success or failure of the method.

## Example

```
<updateversionStatusRequest>
  <id>domain1.pol.pol Library:5:2@69</id>
  <event>checkOut</event>
</updateversionStatusRequest>

<updateversionStatusResponse>
  <part>
    <size>29</size>
    <part>UESDBAoAAAAAON6SjQAAAAAAAAAAAA</part>
  </part>
  <status>
    <code>0</code>
  </status>
</updateversionStatusResponse>
```



# 3 Schema Elements

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

Schemas are object-oriented and inheritance is used frequently. For example, the Appointment element extends CalendarItem, which extends Mail, which extends BoxEntry, which extends ContainerItem, which extends Item.

It is a good idea to gain a basic understanding of the elements in the `types.xsd` file and how they interrelate. The following sections don't walk the inheritance tree. It is up to you as the developer to follow the inheritance tree to its logical leaf nodes.

Each of the following elements has a syntax section with a copy of the element from the schema. The copy of the element has been simplified for readability. Consult the schemas directly for the complete schema definition.

- ♦ [“Simple Objects” on page 270](#)
- ♦ [“Complex Objects” on page 291](#)

# Simple Objects

The following simple objects can be referenced from the [Complex Objects](#) that are listed in the next section:

- ♦ [“acceptLevel” on page 271](#)
- ♦ [“code” on page 272](#)
- ♦ [“description” on page 273](#)
- ♦ [“displayName” on page 274](#)
- ♦ [“e-mail” on page 275](#)
- ♦ [“endDate” on page 276](#)
- ♦ [“gwTrace” on page 277](#)
- ♦ [“id” on page 278](#)
- ♦ [“modified” on page 279](#)
- ♦ [“name” on page 280](#)
- ♦ [“place” on page 281](#)
- ♦ [“recurrenceKey” on page 282](#)
- ♦ [“rights” on page 283](#)
- ♦ [“sequence” on page 284](#)
- ♦ [“session” on page 285](#)
- ♦ [“sid” on page 286](#)
- ♦ [“startDate” on page 287](#)
- ♦ [“subject” on page 288](#)
- ♦ [“uuid” on page 289](#)
- ♦ [“version” on page 290](#)

## **acceptLevel**

Specifies how an appointment shows in a busy search.

### **Syntax**

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

## **code**

Specifies the status of a method response. A return value of 0 indicates the method was successfully executed. If the value is greater than 0, an error occurred during the execution of the method. The description explains the error condition.

### **Syntax**

```
<code type="int"/>
```



## description

Explains the error condition. For more information, see the [code](#) element.

### Syntax

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

## displayName

Specifies the displayable (human readable) name of an object.

### Syntax

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

## **e-mail**

Specifies the email address of an object.

### **Syntax**

```
<e-mail type="string"/>
```

## endDate

Specifies the end date and time of an appointment.

### Syntax

```
<endDate type="definitiondateTime">
```

## gwTrace

If the gwTrace element isTrue, the method request and response is written to a log file in the Post Office `wpcout\ofs` directory. The file name has the date followed by XML (such as `0925xml.001`). The gwTrace value is passed in the SOAP header.

### Syntax

```
<gwTrace type="boolean"/>
```

## id

Specifies the unique identifier of an object.

### Remarks

The ID has two sections:

- ♦ Everything before the first at symbol (@) uniquely identifies an object in GroupWise.
- ♦ Everything after the @ symbol controls access to the item.

### Syntax

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

## modified

Specifies when the object was last modified.

### Syntax

```
<modified type="definitiondateTime"/>
```

## **name**

Specifies the name of an object.

### **Syntax**

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



## place

Specifies a location of an appointment.

### Syntax

```
<place type="xs:string"/>
```

## recurrenceKey

Specifies a common identifier that links recurring items.

### Syntax

```
<recurrenceKey type="unsignedInt"/>
```

## rights

Specifies a user's rights to a specific object. It is an access control list to an object. The boolean rights are read, add, edit, delete, share, and manage.

### Syntax

```
<rights type="Rights"/>
```

## sequence

Specifies the sequence order of an object in a list. For example, the sequence number of a folder in a list of folders is its location or order in the list.

### Syntax

```
<sequence type="int" />
```

## session

Is returned in the response to [loginRequest](#). It is required by all other methods during the user's session and identifies the user that is logged into GroupWise. The session key is passed using the SOAP header.

### Syntax

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

## sid

A short identifier for an item. The sid is not persistent if a user is moved to a different post office.  
**--For GroupWise 8.0 HP1 and later.**

### Syntax

```
<sid type="xs:unsignedInt"/>
```

## startDate

Specifies the starting date of an appointment.

### Syntax

```
<startDate type="dateTime" />
```

## subject

Specifies the subject of an item.

### Syntax

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



## uuid

Specifies the unique identifier of an object.

### Syntax

```
<uuid type="UUID" />
```

## version

Specifies the version of an object.

### Syntax

```
<version type="unsignedInt" />
```

# Complex Objects

The following complex objects can reference any of the [Simple Objects](#) listed in the preceding section:

- ♦ [“AcceptLevel”](#) on page 298
- ♦ [“AccessControlListEntry”](#) on page 299
- ♦ [“AccessControlList”](#) on page 300
- ♦ [“AccessMiscRight”](#) on page 301
- ♦ [“AccessRight”](#) on page 302
- ♦ [“AccessRightChanges”](#) on page 303
- ♦ [“AccessRightEntry”](#) on page 304
- ♦ [“AccessRightList”](#) on page 305
- ♦ [“Accounts”](#) on page 306
- ♦ [“AccountFlags”](#) on page 307
- ♦ [“AddressBook”](#) on page 309
- ♦ [“AddressBookItem”](#) on page 310
- ♦ [“AddressBookList”](#) on page 312
- ♦ [“AdvancedInfo”](#) on page 313
- ♦ [“AgeAction”](#) on page 315
- ♦ [“Alarm”](#) on page 316
- ♦ [“Appointment”](#) on page 317
- ♦ [“AppointmentConflict”](#) on page 318
- ♦ [“AttachmentFlags”](#) on page 319
- ♦ [“AttachmentID”](#) on page 320
- ♦ [“AttachmentInfo”](#) on page 321
- ♦ [“AttachmentItemInfo”](#) on page 322
- ♦ [“Authentication”](#) on page 324
- ♦ [“BoxEntry”](#) on page 325
- ♦ [“CalendarFolderAttribute”](#) on page 327
- ♦ [“CalendarFolderFlags”](#) on page 328
- ♦ [“CalendarItem”](#) on page 329
- ♦ [“CalendarPublish”](#) on page 330
- ♦ [“CalendarPublishRange”](#) on page 331
- ♦ [“CalendarViewType”](#) on page 332
- ♦ [“CalHost”](#) on page 333
- ♦ [“Category”](#) on page 334
- ♦ [“CategoryFlags”](#) on page 335
- ♦ [“CategoryList”](#) on page 336

- ♦ “CategoryRefList” on page 337
- ♦ “ChecklistInfo” on page 338
- ♦ “ColumnSettings” on page 339
- ♦ “ColumnType” on page 340
- ♦ “CommentStatus” on page 341
- ♦ “Contact” on page 342
- ♦ “ContactFolder” on page 344
- ♦ “ContactNote” on page 345
- ♦ “ContactNotes” on page 346
- ♦ “ContactRefList” on page 347
- ♦ “ContactType” on page 348
- ♦ “ContainerItem” on page 349
- ♦ “ContainerRef” on page 350
- ♦ “CursorSeek” on page 351
- ♦ “Custom” on page 352
- ♦ “CustomList” on page 353
- ♦ “CustomType” on page 354
- ♦ “Day” on page 355
- ♦ “DayOfMonth” on page 356
- ♦ “DayOfMonthList” on page 357
- ♦ “DayOfWeek” on page 358
- ♦ “DayOfYear” on page 359
- ♦ “DayOfYearList” on page 360
- ♦ “DayOfYearWeek” on page 361
- ♦ “DaysOfYearWeekList” on page 362
- ♦ “DelegatedStatus” on page 363
- ♦ “DelegateeStatus” on page 364
- ♦ “DeltaInfo” on page 365
- ♦ “DiskSpaceUsage” on page 366
- ♦ “DisplayFlags” on page 367
- ♦ “DisplayFolderType” on page 369
- ♦ “DisplaySettings” on page 370
- ♦ “DisplaySettingsList” on page 371
- ♦ “DisplaySettingsType” on page 372
- ♦ “Distribution” on page 373
- ♦ “DistributionType” on page 374
- ♦ “Document” on page 375
- ♦ “DocumentRef” on page 377

- ♦ “DocumentType” on page 379
- ♦ “DocumentTypeList” on page 380
- ♦ “EmailAddressList” on page 381
- ♦ “Execution” on page 382
- ♦ “External” on page 383
- ♦ “Filter” on page 384
- ♦ “FilterDate” on page 385
- ♦ “FilterElement” on page 386
- ♦ “FilterEntry” on page 387
- ♦ “FilterGroup” on page 388
- ♦ “FilterOp” on page 389
- ♦ “Folder” on page 397
- ♦ “FolderACL” on page 399
- ♦ “FolderACLEntry” on page 400
- ♦ “FolderACLStatus” on page 401
- ♦ “FolderDisplaySettings” on page 402
- ♦ “FolderFlags” on page 403
- ♦ “FolderList” on page 404
- ♦ “FolderType” on page 405
- ♦ “FreeBusyBlock” on page 408
- ♦ “FreeBusyBlockList” on page 409
- ♦ “FreeBusyInfo” on page 410
- ♦ “FreeBusyInfoList” on page 411
- ♦ “FreeBusyStats” on page 412
- ♦ “FreeBusyUserList” on page 413
- ♦ “Frequency” on page 414
- ♦ “From” on page 415
- ♦ “FullName” on page 416
- ♦ “GeneralAccount” on page 417
- ♦ “GMTOffset” on page 418
- ♦ “Group” on page 419
- ♦ “GroupMember” on page 420
- ♦ “GroupMemberList” on page 421
- ♦ “Header” on page 422
- ♦ “Host” on page 423
- ♦ “Hour” on page 424
- ♦ “HTMLImages” on page 425
- ♦ “ImAddress” on page 426

- ♦ [“ImAddressList” on page 427](#)
- ♦ [“IMAP” on page 428](#)
- ♦ [“IMAPFolder” on page 429](#)
- ♦ [“Item” on page 430](#)
- ♦ [“ItemChanges” on page 431](#)
- ♦ [“ItemClass” on page 432](#)
- ♦ [“ItemList” on page 433](#)
- ♦ [“ItemOptions” on page 434](#)
- ♦ [“ItemOptionsPriority” on page 435](#)
- ♦ [“ItemRef” on page 436](#)
- ♦ [“ItemRefList” on page 437](#)
- ♦ [“Items” on page 438](#)
- ♦ [“ItemSecurity” on page 439](#)
- ♦ [“ItemSource” on page 440](#)
- ♦ [“ItemSourceList” on page 441](#)
- ♦ [“ItemStatus” on page 442](#)
- ♦ [“ItemThreading” on page 443](#)
- ♦ [“JunkEntry” on page 444](#)
- ♦ [“JunkHandlingListType” on page 445](#)
- ♦ [“JunkMatchType” on page 446](#)
- ♦ [“Library” on page 447](#)
- ♦ [“LibraryList” on page 448](#)
- ♦ [“LinkInfo” on page 449](#)
- ♦ [“LinkType” on page 450](#)
- ♦ [“Mail” on page 451](#)
- ♦ [“MessageBody” on page 454](#)
- ♦ [“MessagePart” on page 455](#)
- ♦ [“MessageType” on page 456](#)
- ♦ [“MessageTypeList” on page 457](#)
- ♦ [“Minute” on page 458](#)
- ♦ [“ModifyItem” on page 459](#)
- ♦ [“Month” on page 460](#)
- ♦ [“MonthList” on page 461](#)
- ♦ [“MoveItem” on page 462](#)
- ♦ [“NameAndEmail” on page 463](#)
- ♦ [“Nickname” on page 464](#)
- ♦ [“NNTP” on page 465](#)
- ♦ [“NNTPFolder” on page 466](#)

- ♦ “NNTPSets” on page 467
- ♦ “Note” on page 468
- ♦ “NotificationType” on page 469
- ♦ “NotifyEntry” on page 470
- ♦ “NotifyEntryChanges” on page 471
- ♦ “NotifyList” on page 472
- ♦ “OccurrenceType” on page 473
- ♦ “OfficeInfo” on page 474
- ♦ “Organization” on page 475
- ♦ “Owner” on page 476
- ♦ “PanelDisplaySettings” on page 477
- ♦ “PanelList” on page 479
- ♦ “PersonalInfo” on page 480
- ♦ “PhoneFlags” on page 482
- ♦ “PhoneList” on page 483
- ♦ “PhoneMessage” on page 484
- ♦ “PhoneNumber” on page 485
- ♦ “PhoneNumberType” on page 486
- ♦ “PictureData” on page 487
- ♦ “PlainText” on page 488
- ♦ “PollSettings” on page 489
- ♦ “PostalAddress” on page 490
- ♦ “PostalAddressList” on page 491
- ♦ “PostalAddressType” on page 492
- ♦ “ProblemEntry” on page 493
- ♦ “ProblemList” on page 494
- ♦ “Proxy” on page 495
- ♦ “ProxyFolder” on page 496
- ♦ “ProxyList” on page 497
- ♦ “Query” on page 498
- ♦ “QueryFolder” on page 499
- ♦ “QueryTarget” on page 500
- ♦ “QueryTargetList” on page 501
- ♦ “RangeDays” on page 502
- ♦ “Recipient” on page 503
- ♦ “RecipientList” on page 504
- ♦ “RecipientStatus” on page 505
- ♦ “RecipientType” on page 508

- ♦ “RecurrenceDateType” on page 509
- ♦ “RecurrenceRule” on page 510
- ♦ “ReferenceInfo” on page 512
- ♦ “Resource” on page 513
- ♦ “RetractType” on page 514
- ♦ “ReturnNotification” on page 515
- ♦ “ReturnNotificationOptions” on page 516
- ♦ “Rights” on page 517
- ♦ “Rule” on page 518
- ♦ “RuleAction” on page 519
- ♦ “RuleActionList” on page 520
- ♦ “RuleActionType” on page 521
- ♦ “RuleList” on page 522
- ♦ “SendOptions” on page 523
- ♦ “SendOptionsRequestReply” on page 524
- ♦ “SendTotals” on page 525
- ♦ “Server” on page 526
- ♦ “ServerFlags” on page 527
- ♦ “Settings” on page 528
- ♦ “SettingsGroup” on page 529
- ♦ “SettingsList” on page 530
- ♦ “SharedBook” on page 531
- ♦ “SharedFolder” on page 532
- ♦ “SharedFolderNotification” on page 533
- ♦ “SharedNotification” on page 534
- ♦ “Signature” on page 535
- ♦ “SignatureData” on page 536
- ♦ “Signatures” on page 537
- ♦ “SignatureSettings” on page 538
- ♦ “SignatureType” on page 539
- ♦ “SMimeOperation” on page 540
- ♦ “Sort” on page 541
- ♦ “SortType” on page 542
- ♦ “SourceType” on page 543
- ♦ “Status” on page 544
- ♦ “StatusTracking” on page 545
- ♦ “StatusTrackingFlags” on page 546
- ♦ “StatusTrackingOptions” on page 547



- ♦ “Subscribe” on page 548
- ♦ “SystemFolder” on page 549
- ♦ “Task” on page 550
- ♦ “TextLines” on page 551
- ♦ “Timezone” on page 552
- ♦ “TimezoneComponent” on page 553
- ♦ “TimezoneList” on page 554
- ♦ “TransferFailedStatus” on page 555
- ♦ “TrustedApplication” on page 556
- ♦ “uid” on page 557
- ♦ “UserContactFolder” on page 558
- ♦ “UserInfo” on page 559
- ♦ “UserList” on page 561
- ♦ “UUID” on page 562
- ♦ “VacationRule” on page 563
- ♦ “Version” on page 564
- ♦ “VersionEvent” on page 566
- ♦ “VersionEventType” on page 567
- ♦ “VersionStatus” on page 568
- ♦ “View” on page 569
- ♦ “ViewType” on page 570
- ♦ “Visibility” on page 571
- ♦ “WebAccessSettings” on page 572
- ♦ “WeekDay” on page 573

# AcceptLevel

Specifies how appointments show up in free/busy searches. Every appointment in a user's calendar has an acceptLevel.

## Syntax

```
<AcceptLevel>
<enumeration value="Free" />
<enumeration value="Tentative" />
<enumeration value="Busy" />
<enumeration value="OutOfOffice" />
<enumeration value="NoDisplay" />
```

## Definitions

AcceptLevel can have the following values:

---

Free	Show appointments as Free.
Tentative	Show appointments as Tentative. Might be busy.
Busy	Show appointments as Busy.
OutOfOffice	Show appointments as Out of Office.
NoDisplay	Item is not visually displayed in any client.

---

# AccessControlListEntry

Identifies the rights a user has to an object (such as a folder).

## Syntax

```
<AccessControlListEntry>  
<extension base="NameAndEmail">  
<ref="rights"/>
```

# AccessControlList

Specifies a list of [AccessControlListEntry](#) elements.

## Syntax

```
<AccessControlList>  
<entry type="AccessControlListEntry" minOccurs="0" maxOccurs="unbounded"/>
```

# AccessMiscRight

Defines miscellaneous rights that a proxied user has to a mailbox.

## Syntax

```
<AccessMiscRight>  
<alarms type="boolean"/>  
<notify type="boolean"/>  
<readHidden type="boolean"/>  
<setup type="boolean"/>
```

## Definitions

### alarms

Specifies if a proxied user can receive alarms based on appointments in the mailbox.

### notify

Specifies if a proxied user can receive notifications for incoming items in the mailbox.

### readHidden

Specifies if a proxied user can read items that are marked private.

### setup

Specifies if a proxied user can modify your settings, rules, and folders.

# AccessRight

Defines the access rights to mailbox items.

## Syntax

```
<AccessRight>  
<read type="boolean"/>  
<write type="boolean"/>
```

## Definitions

### read

Specifies if a user can read the item type.

### write

Specifies if a user can write the item type.

# AccessRightChanges

Modifies the [AccessRightEntry](#) elements.

## Syntax

```
<AccessRightChanges>  
<add type="AccessRightEntry"/>  
<delete type="AccessRightEntry"/>  
<update type="AccessRightEntry"/>
```

## Definitions

### add

Adds a new user to a user's proxy access list.

### delete

Deletes an existing user in the user's proxy access list.

### update

Updates an existing user in the user's proxy access list.

# AccessRightEntry

Specifies a user that has proxy rights to the mailbox. It also identifies the specific rights the user has to the mailbox.

## Syntax

```
<AccessRightEntry>
  <NameAndEmail>
  <ref="id">
  <appointment type="AccessRight"/>
  <mail type="AccessRight"/>
  <misc type="AccessMiscRight"/>
  <note type="AccessRight"/>
  <task type="AccessRight"/>
```

## Definitions

### id

Specifies the rights entry.

### appointment

Specifies the rights to appointments.

### mail

Specifies the rights to mail items.

### misc

Specifies the miscellaneous rights to the mailbox.

### note

Specifies the rights to notes.

### task

Specifies the rights to tasks.



# AccessRightList

Contains a list of users that have access rights to the mailbox.

## Syntax

```
<AccessRightList>  
<entry type="AccessRightEntry" minOccurs="0" maxOccurs="unbounded"/>
```

# Accounts

Specifies an Account.

## Syntax

```
<Account>  
<extention base="Item">  
<from type="From"/>  
<serverTimeout type="int"/>  
<flags type="AccountFlags"/>
```

## Definitions

### Item

Extend the Item Object.

### from

Identifies the from information defined on the account.

### serverTimeout

Identifies the time in minutes for the server to respond.

### flags

Identifies all the account settings.

# AccountFlags

Specifies Account settings.

## Syntax

```
<addSignature type="xs:boolean"/>
<addVCard type="xs:boolean"/>
<deleteFromTrash type="xs:boolean"/>
<downloadExternalBodies type="xs:boolean"/>
<downloadHeadersOnly type="xs:boolean"/>
<downloadNewOnly type="xs:boolean"/>
<expandThreads type="xs:boolean"/>
<lifeDelete type="xs:boolean"/>
<memberOfFullSync type="xs:boolean"/>
<neverDownloadOldMessages type="xs:boolean"/>
<pollAtStartup type="xs:boolean"/>
<showSubscribedFolder type="xs:boolean"/>
<syncWhenFoldersSelected type="xs:boolean"/>
<watchMyThreads type="xs:boolean"/>
```

## Definitions

### **addSignature**

Specifies whether to add the existing signature.

### **addVCard**

Specified whether to add the existing vCard.

### **deleteFromTrash**

Specifies whether to remove items from trash as they deleted.

### **downloadExternalBodies**

Specifies whether to download external voice mail attachment if the IMAP account is a telephony IMAP server.

### **downloadHeadersOnly**

Specifies whether to download only the headers in the message.

### **downloadNewOnly**

Specifies whether to download only new messages.

### **expandThreads**

Specifies whether to expand NNTP threads when they are downloaded.

### **lifeDelete**

Specifies to remove old messages that are so many days old.

### **memberOfFullSync**

Specifies whether to include this account when doing Send/Retrieve on all Marked Accounts.

**neverDownloadOldMessages**

Specifies never to download old NTP messages.

**pollAtStartup**

Specify to Send/Retrieve when the folder is selected.

**showSubscribedFolders**

Specified whether to display only subscribed folders.

**syncWhenFolderIsSelected**

Specified whether to sync messages when folder is selected.

**watchMyThreads**

Specifies whether to watch for threads with posting for me.

# AddressBook

Specifies an address book.

## Syntax

```
<AddressBook>
<extension base="Item">
<ref="description"/>
<isPersonal type="boolean"/>
<isFrequentContacts type="boolean"/>
<count type="int"/>
</AddressBook>
```

## Definitions

### Item

Inherits from the [Item](#) element.

### description

Specifies the description of the address book.

### isPersonal

Specifies if this is a personal address book.

### isFrequentContacts

Specifies if this is the Frequent Contacts Address Book, which is considered to be a personal address book.

### count

If you put “count” in the view when you get the address book(s), it will return the number of items in the book.

---

**NOTE:** This only works on personal address books.

---

**--For GroupWise 2012 and later.**

# AddressBookItem

Contains the base object for address book items.

## Syntax

```
<AddressBookItem>
<extension base="ContainerItem"/>
<ref="uuid"/>
<comment type="string"/>
<sync type="DeltaSyncType"/>
<domain type="string"/>
<postOffice type="string"/>
<distinguishedName type="string"/>
<userid type="string"/>
<PABGuid/ type="string"/>
<visibility type="Visibility"/>
<attribute name="external" type="boolean"/>
</AddressBookItem>
```

## Definitions

### ContainerItem

Specifies the [ContainerItem](#) element.

### uuid

Specifies the unique identifier for this item.

### comment

Specifies the comment for this address book item.

### sync

Specifies the DeltaSyncType element.

### domain

Specifies the domain of this address book item (available on GroupWise Address Book items).

### postOffice

Specifies the post office of this address book item (available on GroupWise Address Book items).

### distinguishedName

Specifies the distinguishedName of this address book item (available on GroupWise Address Book items).

### userid

Specifies the userid of this address book item.

**PABGuid**

Specifies the PABG ID. Include PABGuid in the view to retrieve this element.

**--For GroupWise 8.0 SP2 and later.**

**visibility**

Specifies the visibility of the item (available on GroupWise Address Book items).

**external**

Attribute that specifies if it is an external item (available on GroupWise Address Book items).

# AddressBookList

Defines the list of address books.

## Syntax

```
<AddressBookList>  
<book type="AddressBook" minOccurs="0" maxOccurs="unbounded"/>
```



# AdvancedInfo

Specifies additional elements on a Contact.

## Syntax

```
<freebusyWebsite type="xs:string"/>
<teamingWebsite type="xs:string"/>
<customerId type="xs:string"/>
<governmentId type="xs:string"/>
<organizationalId type="xs:string"/>
<computerNetworkName type="xs:string"/>
<ftpSite type="xs:string"/>
<mileage type="xs:string"/>
<userField1 type="xs:string"/>
<userField2 type="xs:string"/>
<userField3 type="xs:string"/>
<userField4 type="xs:string"/>
<BlackBerryPIN type="xs:string"/>
```

## Definitions

### **freebusyWebsite**

Specifies a URL where GroupWise will query free/busy information on an external contact. The external contact will provide the URL.

### **teamingWebsite**

Specifies the URL for the teaming website.

### **customerID**

Specifies a customer id.

### **governmentId**

Specifies a government id.

### **organizationalId**

Specifies an organizational id.

### **computerNetworkName**

Specifies the name of a computer network.

### **ftpSite**

Specifies a ftp site.

### **mileage**

Specifies mileage.

### **userField1**

Specifies a user defined field.

**userField2**

Specifies a user defined field.

**userField3**

Specifies a user defined field.

**userField4**

Specifies a user defined field.

**BlackBerryPIN**

Specifies a BlackBerry PIN.

# AgeAction

Specifies the action to perform when a document reaches a specified age.

## Syntax

```
<AgeAction>  
<enumeration value="archive"/>  
<enumeration value="delete"/>  
<enumeration value="retain"/>
```

## Definitions

### archive

Specifies when the document item is archived.

### delete

Specifies when the document item is deleted.

### retain

Specifies when the document item is retained.

# 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](#) 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.

# AppointmentConflict

Determines how to handle appointment scheduling conflicts.

## Syntax

```
<AppointmentConflict>  
<enumeration value="Yes"/>  
<enumeration value="No"/>  
<enumeration value="Ignore"/>
```

## Definitions

### Yes

Specifies that rule actions are executed when there are conflicting appointments.

### No

Specifies that rule actions are executed when there are no conflicting appointments.

### Ignore

Specifies to ignore conflicting appointments.

# AttachmentFlags

Specifies attachment flags. The flags work only if you get the entire attachment. This means you need to specify an offset of 0 and a length of -1 in the `getAttachmentRequest`. GroupWise clients will create the hash on new items. If a hash has already been created, it will be 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 64-bits. The hash string is a 16 byte hex string.

## Syntax

```
<AttachmentFlags>
  <enumeration value="ComputeHash" />
  <enumeration value="Deflate" />
  <enumeration value="ReturnData" />
  <enumeration value="ReturnPath" />
  <enumeration value="Mime" />
</AttachmentFlags>
```

## Definitions

### ComputerHash

Specify the `ComputerHash` flag to create a hash value for an attachment.

### Deflate

Reserved. Not implemented.

### ReturnData

Specify the `ReturnData` flag if you want the attachment data returned.

### ReturnPath

Specify the `ReturnPath` flag if you want the full path to the attachment data blob returned. This is used to find the path to the data blob, usually to restore the data blob back. If this flag is specified, the other flags are ignored.

**--For GroupWise 8.0 SP1 and later.**

# 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 (for example, mail, appointment, task, note, etc.). 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"/>
```

## Definitions

### attachment

Specifies 0 or more [AttachmentItemInfo](#) elements.

# AttachmentItemInfo

Describes an attachment.

## Syntax

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

## Definitions

### id

Specifies the attachment.

### name

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

### charset

Specifies the character set used in the attachment.

--For GroupWise 2012 and later.

### 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.

**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.

**isPersonal**

This value is true if the attachment is a personal attachment on a received item.

**hash**

A 16-byte hex hash value. See [AttachmentFlags](#) for more information.

# Authentication

Contains the base object for plainText, Proxy, and Trusted Application login.

## Syntax

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

## Definitions

### **userName**

The user name used during authentication to GroupWise.

# BoxEntry

Is an abstraction layer that is used to contain information that is common to all items that can be in a GroupWise mailbox (not 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 above except that the format is different. messageId can help applications match up items retrieved with the IMAP and SOAP protocols.

### clientMessageId

This is the same id that you will see when you open an item in the GroupWise Window 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 up to the recipients to abide by the sender's request.

**comment**

Specifies the comment on a item.

# CalendarFolderAttribute

Specifies the attributes for a calendar. Multiple calendar support is available.

## Syntax

```
<CalendarFolderAttribute>  
  <flags type="CalendarFolderFlags" />  
  <color type="unsignedInt" />  
  <calSequence type="int" />  
  <publish type="CalendarPublish" />  
</CalendarFolderAttribute>
```

## Definitions

### flags

Specifies the calendar folder flags for this calendar.

### color

Specifies the color of the calendar folder.

### calSequence

The sequence order of this calendar folder in the list of calendar folders.

### publish

Calendar publish settings.

In GroupWise 8.0, personal calendars can be published to a Calendar Host. This is deprecated in GroupWise 2012. Uses the [CalendarPublish](#) settings instead.

# CalendarFolderFlags

Specifies the attributes for a calendar. Multiple calendar support is available and each calendar can have a different set of attributes. Users can set whether a particular calendar appears in the overall calendar view.

## Syntax

```
<CalendarFolderFlags>  
<enumeration value="ShowInList" />  
<enumeration value="DontIncludeContent" />  
<enumeration value="Publish" />
```

## Definitions

### ShowInList

Specifies what calendar items to show in the calendar view.

### DontIncludeContent

Specifies what calendar items not to show in the aggregate calendar view.

### Publish

Personal calendars can be published to a Calendar Host.



# CalendarItem

Is 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](#).

### rdate

Specifies a list of recurring dates.

### rrule

Specifies a recurrence rule.

### exdate

Specifies a list of dates to exclude. For example, if you wanted 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 to accept all recurring items).

### iCalId

Specifies the iCalendar (RFC2445) UID.

# CalendarPublish

Controls the publishing of a calendar. --For GroupWise 2012 and later.

## Syntax

```
<CalendarPublish>
  <enabled type="boolean"/>
  <range type="CalendarPublishRange"/>
  <relative type="RangeDays"/>
  <includePrivate type="boolean"/>
</CalendarPublish>
```

## Definitions

### enabled

Whether or not the calendar is enabled to be published.

### range

Range of days to publish.

### relative

Relative range of days to publish.

### includePrivate

Whether or not to publish private items.

# CalendarPublishRange

Controls the range of dates to publish in a calendar. --For GroupWise 2012 and later.

## Syntax

```
<CalendarPublishRange>  
  <enumeration value="EntireCalendar"/>  
  <enumeration value="Relative"/>  
</CalendarPublishRange>
```

## Definitions

### EntireCalendar

Publish the entire calendar.

### Relative

Publish a range of days.

# CalendarViewType

Identifies the calendar view type. Meaning , the calendar view is set to day, month, multiUser, week, or year.

## Syntax

```
<CalendarViewType>  
<enumeration value="day" />  
<enumeration value="month" />  
<enumeration value="multiUser" />  
<enumeration value="week" />  
<enumeration value="year" />
```

# CalHost

A definition for a calendar publishing host. --For GroupWise 2012 and later.

## Syntax

```
<CalHost>  
  <extension base="Item"/>  
  <ref name="description"/>  
  <host type="Host"/>  
  <URL type="string"/>  
</CalHost>
```

## Definitions

### description

Specifies descriptive text for the calendar host.

### host

Specifies the definition of the host.

### URL

Specifies the URL to use in publishing the calendar data.

# Category

Contains personalization that user can apply to items.

## Syntax

```
<Category>
  <extension base="Item">
    <type type="CategoryType"/>
    <color type="unsignedInt"/>
    <background type="unsignedInt"/>
    <flags type="CategoryFlags"/>
  </Category>
```

## Definitions

### type

Specifies a list of predefined categories: Personal, Follow-up, Urgent, or LowPriority and Normal.

### color

Specifies the color to display with a category. The color is in Windows RGB format.

### background

Specifies the background color for a defined category.

### flags

Specifies special flags on categories.

# CategoryFlags

Contains the flags used to mark categories.

## Syntax

```
<CategoryFlags>  
  <notInMasterList type="boolean" />  
</CategoryFlags>
```

## Definitions

### notInMasterList

Specifies that this category is not in the master list (ie. this category is hidden).

## CategoryList

Contains a list of categories. Each GroupWise user can add, delete and modify categories. Each category has an ID that uniquely identifies it. [getCategoryListRequest](#) returns the list of categories for a user. GroupWise allows for a primary category. If an item has one or more categories, it must specify one category as the primary category.

### Syntax

```
<CategoryList>  
<category type="Category" minOccurs="0" maxOccurs="unbounded" />
```



# 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.

# ChecklistInfo

Returns checklist information on an item. In GroupWise 8.0, the checklist folder was renamed the Tasklist folder.

## Syntax

```
<ChecklistInfo>
<sequence type="unsignedInt" />
<dueDate value="dateTime" />
<percentComplete type="int" />
<completed type="dateTime" />
<thread type="string" />
<alarm type="Alarm" />
<alarmDate type="AlarmDate" />
<taskTimeOffset type="unsingedInt" />s
```

## Definitions

### sequence

Specifies the order of the item in the checklist.

### dueDate

Specifies the date the item is due.

### percentComplete

Tasklist items can be marked with a percent complete value of 0-100%.

### completed

The completed field is the date and time in UTC that the item was marked complete. If an item is marked complete, the percentComplete is automatically changed to 100%.

### thread

This read-only field specifies the indentation level for sub-tasks. For example, if you have two values separated by a colon (":")(for example. 13D5:13FF), it means the item is indented one level. If there are two colons, it means the indentation is two levels.

### alarm

### alarmDate

### taskTimeOffset

Specifies the time offset that the task is due on the due date. The offset is specified from midnight and is entered in seconds.

--For GroupWise 2014 R2 SP1 and later.

# ColumnSettings

Specifies the column names and size information for folders.

## Syntax

```
<ColumnSettings>  
<type type="ColumnType" />  
<sort type="Sort" />  
<headers type="Header" />
```

## Definitions

### ColumnType

Specifies the columnType for a folder.

### Sort

Specifies the sort order for a folder - ascending or descending.

### Header

Specifies the name and size of column headers.

# ColumnType

The Windows and Linux clients can have different column settings. For example, the number of columns and sort order can be different. The ColumnType specifies which client the setting apply too.

## Syntax

```
<ColumnType>  
<enumeration value="Windows" />  
<enumeration value="Java" />
```

# CommentStatus

Contains the status of a comment. Some item actions request a comment. For example, when accepting an appointment, a user can provide the originator a comment.

## Syntax

```
<CommentStatus>  
<extension base="dateTime"  
<attribute name="comment" type="string"/>
```

## Definitions

### dateTime

Specifies the date and time that the action occurred. For example, when accepting or declining an appointment.

### comment

Specifies the comment associated with the action.

# Contact

Describes a person that you communicate with.

## Syntax

```
<Contact>
  <extension base="AddressBookItem">
    <fullName type="FullName"/>
    <e-mailList type="EmailAddressList"/>
    <imList type="ImAddressList"/>
    <phoneList type="PhoneList"/>
    <addressList type="PostalAddressList"/>
    <officeInfo type="OfficeInfo"/>
    <personalInfo type="PersonalInfo"/>
    <referenceInfo type="ReferenceInfo"/>
    <advancedInfo type="AdvancedInfo"/>
    <contactNotes type="ContactNotes"/>
    <vcard type="AttachmentItemInfo"/>
  </Contact>
```

## Definitions

### **fullName**

Specifies the parts of the contact's name.

### **mailList**

Specifies a list of the contact's email addresses.

### **imList**

Specifies a list of the contact's instant messaging addresses.

### **phoneList**

Specifies a list of the contact's phone numbers.

### **addressList**

Specifies the contact's home and business mailing addresses.

### **officeInfo**

Specifies miscellaneous office information.

### **personalInfo**

Specifies miscellaneous personal information.

### **referenceInfo**

Specifies the last date and number of times the contact was referenced.

### **advancedInfo**

Additional fields supported on a contact.

**contactNotes**

A set of notes added to a contact.

**vcard**

Used to import a vcard as a contact. The data from the vCard is parsed and a Contact is created.

--For GroupWise 2012 and later.

# ContactFolder

Represents the GroupWise Contacts folder. The contacts folder can reference any of the user's personal address books.

## Syntax

```
<ContactFolder>  
<extension base="SystemFolder"  
<addressBook type="uid" />
```

## Definitions

### addressBook

Specifies the ID of the personal address book to reference in the contact folder.



# ContactNote

Specifies a ContactNote. For more information on how to create, modify, and delete contactNotes, see [ContactNotes on a Contact](#).

## Syntax

```
<ContactNote>  
<id type="uid">  
<created type="dateTime">  
<attachment type="AttachmentItemInfo">
```

## Definitions

### id

The id of the ContactNote.

### created

The date the ContactNote was created.

### attachment

The AttachmentItemInfo that describes the ContactNote.

## ContactNotes

contactNotes expand on the comment field in AddressBookItem. Contacts can have a list of contactNotes. Each note has a create date and text. For more information on how to create, modify, and delete contactNotes, see [ContactNotes on a Contact](#).

### Syntax

```
<ContactNotes>  
<contactNote type="ContactNote" minOccurs="0" maxOccurs="unbounded" />
```

## ContactRefList

A list of contacts can be added to received items. In GroupWise windows client, open a received item and click on the personalize tab. Notice that you can add a list of contacts to this item. This personalization is useful to link specific items to a contact. To see this linkage, open a contact and view the history tab.

### Syntax

```
<ContactRefList>  
<contact type="uid" minOccurs="0" maxOccurs="unbounded"/>
```

### Definitions

#### contact

Unique id for a contact.

# ContactType

Identifies the item type in a group: contact, group, resource, or organization.

## Syntax

```
<ContactType>  
<enumeration value="Contact" /  
<enumeration value="Group" /  
<enumeration value="Resource" /  
<enumeration value="Organization" /
```

# 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. See [ContactRefList](#) for more information.

# ContainerRef

Identifies a container.

## Syntax

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

## Definitions

### deleted

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

### sid

Short id of the container.

**--For GroupWise 8.0 HP1 and later.**

# CursorSeek

Identifies where to position the cursor for the next read.

## Syntax

```
<CursorSeek>  
<enumeration value="current"/>  
<enumeration value="start"/>  
<enumeration value="end"/>
```

## Definitions

### current

Specifies to use the current position of the cursor as the starting point.

### start

Specifies to use the start of the item list as the starting point of the read.

### end

Specifies to use the end of the item list as the starting point of the read.

# 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 the Tools > Options > Send > Mail > Return Notification. It also applies Appointments, Tasks, and Reminder Notes.

### locked

Specifies if the value cannot be changed.

### adminOnly

AdminOnly will be 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"/>
```

# DayOfMonth

Restricts a day of the month to be between -31 and 31.

## Syntax

```
<DayOfMonth>  
  <restriction base="byte">  
    <minInclusive value="-31"/>  
    <maxInclusive value="31"/>  
</DayOfMonth>
```

# DayOfMonthList

Contains a list of [DayOfMonth](#) elements.

## Syntax

```
<DayOfMonthList>  
<day type="DayOfMonth" minOccurs="0" maxOccurs="unbounded"/>
```

# DayOfWeek

Defines a start or end day and the week in the month. For example, <dayOfWeek occurrence="First">Sunday</dayOfWeek> reads 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.

# DayOfYear

Restricts a day in a year to be between -366 to 366.

## Syntax

```
<DayOfYear>  
  <restriction base="short">  
    <minInclusive value="-366"/>  
    <maxInclusive value="366"/>  
</DayOfYear>
```

# DayOfYearList

Specifies a list of [DayOfYear](#) elements.

## Syntax

```
<DayOfYearList>  
<element name="day" type="DayOfYear" minOccurs="0" maxOccurs="unbounded"/>
```



# DayOfYearWeek

Contains the day of the week—[DayOfWeek](#)—with an occurrence during the week in the year.

## Syntax

```
<DayOfYearWeek>  
<extension base="WeekDay">  
<attribute name="occurrence" type="WeekOfYear"/>
```

## Definitions

### WeekDay

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

### WeekOfYear

Specifies the week in the year: -53 to 53.

# DaysOfYearWeekList

Contains a list of [DayOfYearWeek](#) elements.

## Syntax

```
<DaysOfYearWeekList>  
<day type="DayOfYearWeek" minOccurs="0" maxOccurs="unbounded"/>
```

# DelegatedStatus

As part of [RecipientStatus](#), tracks when an appointment is delegated to another user. DelegatedStatus is tracked by the originator of the appointment.

## Syntax

```
<DelegatedStatus>  
<extension base="CommentStatus">  
<attribute name="userid" type="string"/>
```

## Definitions

### userid

Specifies the email address of the person that the appointment was delegated to.

# DelegateeStatus

Tracks the status of items that have been delegated.

## Syntax

```
<DelegateeStatus>  
<attribute name="userid" type="string"/>
```

## Definitions

### userid

Specifies the email address of the delegatee.

# DeltaInfo

Contains pointers or sequence numbers into a dynamic list of GroupWise Address Book changes. A GroupWise administrator can add, delete, and modify existing users, groups, or resources in the GroupWise Address Book. As these GroupWise Address Book values change, the deltaInfo structure grows.

## Syntax

```
<DeltaInfo>
<count type="unsignedInt"/>
<firstSequence type="unsignedInt"/>
<lastSequence type="unsignedInt"/>
<lastTimePORebuild type="unsignedInt"/>
```

## Definitions

### count

Inputs the desired number of items to be returned in one read. 1 indicates to return all the items in the list. Outputs the actual number of items returned.

### firstSequence

Inputs the sequence or position to start the read. Outputs the first valid sequence in the returned list.

### lastSequence

Outputs the last sequence number that was successfully read.

### lastTimePORebuild

Specifies the last time an administrator rebuilt the post office. A post office rebuild resets all the sequence numbers. If a rebuild occurs, a resynchronization is required before new deltas can be applied to a local list.

### DeltaSyncType

Specifies an enumeration of change types to the GroupWise Address Book:

```
<enumeration value="add"/>
<enumeration value="delete"/>
<enumeration value="update"/>
```

# DiskSpaceUsage

Specifies the disk space usage for a user.

## Syntax

```
<DiskSpaceUsage>
  <maxSize" type="unsignedInt"/>
  <threshold" type="unsignedInt"/>
  <used" type="unsignedInt"/>
  <attribute name="inMB" type="boolean"/>
</DiskSpaceUsage>
```

## Definitions

### maxSize

Specifies the maximum size of a mailbox for a user in bytes.

### threshold

Specifies the user threshold to warn users when they are about to run out of disk space. This value is a percentage between 0-100%.

### used

Specifies the used disk space for a user in bytes.

### inMB

Specifies if the values in megabytes instead of bytes.

**--For GroupWise 8.0 SP1 and later.**

# DisplayFlags

Specifies DisplayFlag settings on a folder. To view the settings in the Windows client, select a folder > right click > properties > Display tab.

## Syntax

```
<DisplayFlags>
<hideNonTasklistItems type="boolean"/>
<showQuickViewer type="boolean"/>
<rememberQuickViewer type="boolean"/>
<summary type="boolean"/>
<showGroupLables type="boolean"/>
<showFolderTree type="boolean"/>
<rememberFolderTree type="boolean"/>
<showSimpleFolderTree type="boolean"/>
<showFullFolderTree type="boolean"/>
<messagePreview type="boolean"/>
<hideCompleted type="boolean"/>
<hideCompletedAfterOneDay type="boolean"/>
```

## Definitions

### hideNonTasklistItems

Hide items not in the tasklist.

### showQuickViewer

Show quick view of the selected item.

### rememberQuickViewer

Remember the quick viewer folder selection.

### summary

Do not show the column names. Rather, show a summary of the data.

### showGroupLables

Show a group label in the folder list. For example, items in the folder list are separated by the date. The date is the group label.

### showFolderTree

Show the folder tree.

### rememberFolderTree

Remember the folder tree settings.

### showSimpleFolderTree

Show the full folder tree.

### messagePreview

Shows a preview of the message body under the From and Subject column in the detail view.

**hideCompleted**

Hide completed tasklist items one day after completed.

**hideCompletedAfterOneDay**

Hide completed tasklist items one day after completed.



# DisplayFolderType

Specifies the folder type.

## Syntax

```
<DisplayFolderType>
<enumeration value="Calendar"/>
<enumeration value="Checklist"/>
<enumeration value="Contacts"/>
<enumeration value="ContactsPersonal"/>
<enumeration value="ContactsOffice"/>
<enumeration value="ContactsUserDefined"/>
<enumeration value="DiscussionThread"/>
<enumeration value="DiscussionThreadList"/>
<enumeration value="Documents"/>
<enumeration value="Draft"/>
<enumeration value="Mailbox"/>
<enumeration value="NNTPServer"/>
<enumeration value="Normal"/>
<enumeration value="PanelCalendar"/>
<enumeration value="PanelChecklist"/>
<enumeration value="PanelSummaryCalendar"/>
<enumeration value="PanelUnreadItems"/>
<enumeration value="Query"/>
<enumeration value="QueryResults"/>
<enumeration value="SentItems"/>
<enumeration value="TaskList"/>
<enumeration value="Toolbar"/>
<enumeration value="Trash"/>
<enumeration value="Unknown"/>
<enumeration value="UserDefined"/>
<enumeration value="VersionList"/>
```

# DisplaySettings

Specifies the DisplaySettings on a folder or panel.

## Syntax

```
<DisplaySettings>
<folderType type="DisplayFolderType"/>
<settingType type="DisplaySettingsType"/>
<flags type="DisplayFlags"/>
<types type="string"/>
<source type="string"/>
<view type="ViewType"/>
<columns type="ColumnSettings"/>
<filterDaysForward type="int"/>
<filterDayBackward type="int"/>
```

## Definitions

### folderType

Specifies the folder type. See [DisplayFolderType](#) for more information.

### settingType

Specifies the display settings type. See [DisplaySettingsType](#) for more information.

### flags

Specifies the display flags. See [DisplayFlags](#) for more information.

### types

Specifies the message type. See [MessageType](#) for more information.

### source

Specifies the item source. See [ItemSource](#) for more information.

### columns

Specifies the view type. See [ColumnSettings](#) for more information.

### filterDaysForward

Specifies the number of days forward to get items.

### filterDays Backward

Specifies the number of days backward to get items.

# DisplaySettingsList

Specifies a list of display settings.

## Syntax

```
<DisplaySettingsList>  
<settings type="DisplaySettings"/>
```

# DisplaySettingsType

Specifies the display setting type.

## Syntax

```
<DisplaySettingsType>  
<enumeration value="Folder"/>  
<enumeration value="FolderDefaults"/>  
<enumeration value="Panel"/>  
<enumeration value="PanelTemplate"/>  
<enumeration value="Unknown"/>
```

# 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 displayable string of the To recipients.

### cc

Specifies the displayable string of the CC recipients.

### bc

Specifies the displayable string of the BC recipients.

### recipients

Specifies the collection of recipients.

### sendOptions

Specifies the 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.

# Document

Contains a document.

## Syntax

```
<Document>
  <extension base="ContainerItem">
    <element ref="subject"/>
    <library type="NameAndEmail"/>
    <documentNumber type="unsignedInt"/>
    <documentTypeName type="string"/>
    <author type="NameAndEmail"/>
    <creator type="NameAndEmail"/>
    <officialVersion type="unsignedInt"/>
    <currentVersion type="unsignedInt"/>
    <current type="AccessControlList"/>
    <official type="AccessControlList"/>
    <other type="AccessControlList"/>
    <size type="int"/>
    <filename type="string"/>
  </Document>
```

## Definitions

### subject

Specifies the subject of a document.

### library

Specifies the document library.

### documentNumber

Specifies the document number in the library.

### documentTypeName

Specifies the documentType.

### 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.

### current

Specifies the access control list for the current version.

**official**

Specifies the access control list for the official version.

**other**

Specifies the access control list for all other versions.

**size**

Specifies the size of the document.

**filename**

Specifies the document file name.



# 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 documentType.

### **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.

# DocumentType

Contains the document type. Each document has an associated document type.

## Syntax

```
<DocumentType>  
<sequence>  
<element ref="name"/>  
<life type="unsignedInt"/>  
<maximumVersions type="unsignedInt"/>  
<ageAction type="AgeAction"/>
```

## Definitions

### name

Specifies the name of the document type.

### life

Specifies the document life in days. After the life in days, the ageAction occurs.

### maximumVersions

Specifies the maximum number of versions for the document.

### ageAction

Specifies the action (archive, delete, etc.) that occurs when the life of the document reaches the life value in days.

# DocumentTypeList

Contains a list of [DocumentType](#). Call [getDocumentTypeListRequest](#) to return this list.

## Syntax

```
<DocumentTypeList>  
<item type="DocumentType" minOccurs="0" maxOccurs="unbounded"/>
```

# EmailAddressList

Contains of list of email addresses for a contact.

## Syntax

```
<EmailAddressList>  
<ref="e-mail" minOccurs="0" maxOccurs="unbounded"/>  
<attribute name="primary" type="string"/>
```

## Definitions

### primary

Specifies the default email address.

# Execution

Contains the execution time of a rule.

## Syntax

```
<Execution>
<enumeration value="Startup"/>
<enumeration value="Exit"/>
<enumeration value="New"/>
<enumeration value="FolderOpen"/>
<enumeration value="FolderClose"/>
<enumeration value="FolderNew"/>
<enumeration value="Completed"/>
<enumeration value="User"/>
<enumeration value="Unknown"/>
```

## External

Specifies whether an item (for example, contact) has been synced from another GroupWise system into the address book.

### Syntax

```
<attribute name="external" type="boolean"/>
```

# Filter

Specifies how you want to filter items. For more information see [Filters and Views](#).

## 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. For more information see [Filters and Views](#).

## Syntax

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

# FilterElement

Is the base object for a filter. For more information see [Filters and Views](#).

## Syntax

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

## Definition

### op

Specifies the operation applied to a field and value.

# FilterEntry

Defines a filter. FilterEntry extends [FilterElement](#). For more information see [Filters and Views](#).

## 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

Mask applies to the bitCare FilterOp. It is a bit operator.

# FilterGroup

Groups a collection of [FilterEntry](#) or [FilterGroup](#) elements. FilterGroup uses and, or, or not to group elements. For more information see [Filters and Views](#).

## Syntax

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

# FilterOp

Lists valid operations on a filter. For more information see [Filters and Views](#).

## 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>ne</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>lte</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 Appointment:

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

## fieldEqual

Is 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

Is 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

Is 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**

Is 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**

Is 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**

Is 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**

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

### **bitCare**

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 “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 do not contain "Today".

```

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

```

# Folder

Is the base object for all folders.

## Syntax

```
<Folder>
  <extension base="Item">
    <parent type="uid"/>
    <element ref="description"/>
    <count type="int"/>
    <hasUnread type="boolean"/>
    <unreadCount type="int"/>
    <element ref="sequence"/>
    <calendarAttribute type="CalendarFolderAttribute"/>
    <URL type="string"/>
    <frequency type="int"/>
    <displaySettings type="FolderDisplaySettings"/>
    <displayReference type="uid"/>
    <flags type="FolderFlags"/>
  </Folder>
```

## Definitions

### parent

Specifies the ID of the parent folder.

### description

Specifies the description of the folder.

### count

Specifies the number of items in a folder.

The count is not returned on the Sent Items folder, folders shared to me, and Query folders. The reason for not returning the counts on the above folder types is because we need to read all the items and then discard the results to get the count. The resource cost was deemed to high.

### hasUnread

Specifies if there are unread items in the folder.

### unreadCount

Specifies the number of unread items in a folder.

### sequence

Specifies the location or sequence of a folder in a folder list.

### settings

Is not currently used.

### calendarAttribute

Specifies the calendarAttribute element.

**URL**

Specifies the URL assigned to a folder. URL's can be assigned to a panel.

**frequency**

Specifies the refresh frequency on a folder. Frequency can be assigned on a panel if it applies.

**displaySettings**

Specifies the folder of panel's display settings.

**displayRefernce**

Reference to display settings.

**--For GroupWise 8.0 SP1 and later.**

**flags**

Specifies control flags of the folder.

**--For GroupWise 2012 and later.**

## FolderACL

Contains a list of users and their rights on a folder. For a folder that is shared to other users (sharedByMe), the folderACL lists all the users and their rights to the shared folder. For a folder that is shared to a user (sharedToMe) by another user, the folderACL lists only the user that received the share.

### Syntax

```
<FolderACL>  
<entry type="FolderACLEntry" minOccurs="0" maxOccurs="unbounded"/>
```

# FolderACLEntry

Contains the user and the rights to a shared folder.

## Syntax

```
<FolderACLEntry>  
<extension base="AccessControlListEntry"/>  
<status type="FolderACLStatus"/>
```

## Definitions

### AccessControlListEntry

FolderACLEntry extends [AccessControlListEntry](#).

### status

Specifies the state of the folder shared to this user: accepted, deleted, etc.



## FolderACLStatus

Contains an enumeration of the possible statuses for a shared folder.

### Syntax

```
<simpleType name="FolderACLStatus"
  <restriction base="string"
    <enumeration value="pending"/>
    <enumeration value="accepted"/>
    <enumeration value="deleted"/>
    <enumeration value="opened"/>
    <enumeration value="owner"/>
```

# FolderDisplaySettings

Specifies the Display Settings for information on how to retrieve the settings. See [DisplaySettings](#) for information on how to retrieve the settings.

## Syntax

```
<FolderDisplaySettings>  
<extension base="DisplaySettings">  
<panelColumns type="int"/>  
<panels type="PanelList"/>
```

## Definitions

### panelColumns

Specifies the number of panels columns.

### panels

Specifies the defined panels.

# FolderFlags

Controls flags for a folder.

## Syntax

```
<FolderFlags>  
  <firstUnread type="boolean"/>  
</FolderFlags>
```

## Definitions

### firstUnread

Positions to the first unread item when the folder is selected.

--For GroupWise 2012 and later.

# FolderList

Contains a list of folders.

## Syntax

```
<FolderList>  
<folder type="Folder" minOccurs="0" maxOccurs="unbounded"/>
```

# FolderType

Contains an enumeration of GroupWise system folders.

## Syntax

```
<FolderType>
  <enumeration value="Mailbox"/>
  <enumeration value="SentItems"/>
  <enumeration value="Draft"/>
  <enumeration value="Trash"/>
  <enumeration value="Calendar"/>
  <enumeration value="Contacts"/>
  <enumeration value="Documents"/>
  <enumeration value="Checklist"/>
  <enumeration value="Cabinet"/>
  <enumeration value="Normal"/>
  <enumeration value="NNTPServer"/>
  <enumeration value="NNTPNewsGroup"/>
  <enumeration value="IMAP"/>
  <enumeration value="Query"/>
  <enumeration value="Root"/>
  <enumeration value="JunkMail"/>
  <enumeration value="Notes"/>
  <enumeration value="RSS"/>
  <enumeration value="Subscribe"/>
  <enumeration value="UserContacts"/>
  <enumeration value="MAPIRoot"/>
  <enumeration value="Proxy"/>
  <enumeration value="Teaming"/>
  <enumeration value="TeamingMyTeams"/>
  <enumeration value="TeamingFavorites"/>
</FolderType>
```

## Definitions

### Mailbox

Specifies the Mailbox system folder.

### SentItems

Specifies the SentItems system folder.

### Draft

Specifies the Work In Progress system folder.

### Trash

Specifies the Trash system folder.

### Calendar

Specifies the Calendar system folder.

**Contacts**

Specifies the Contacts system folder.

**Documents**

Specifies the Documents system folder.

**Checklist**

Specifies the Checklist/Tasklist folder.

In GroupWise 8.0, the checklist folder was renamed to Tasklist.

**Cabinet**

Specifies the Cabinet system folder.

**Normal**

Specifies a Normal folder.

**NNTPServer**

Specifies a NNTPServer folder.

**NNTPNewsGroup**

Specifies a NNTPNewsGroup folder.

**IMAP**

Specifies an IMAP folder.

**Query**

Specifies the Root system folder.

**Root**

Specifies the Root system folder.

**JunkMail**

Specifies the JunkMail system folder.

**Notes**

Specifies the Notes system folder.

**RSS**

Specifies a RSS folder.

**Subscribe**

Specifies a subscribed calendar folder.

**UserContacts**

Specifies a Personal Address Book (PAB) folder.

**MAPIRoot**

Specifies the root MAPI folder.

--For GroupWise 2012 and later.

**Proxy**

Specifies the Calender folder of a proxy user.

**--For GroupWise2012 and later.**

**Teaming, TeamingMyTeams, TeamingFavorites**

Specifies Teaming (Open Text Vibe) folders.

**--For GroupWise 2012 and later.**

# FreeBusyBlock

Returns a free/busy block of time.

## Syntax

```
<FreeBusyBlock>  
<ref="startDate"/>  
<ref="endDate"/>  
<ref="acceptLevel"/>  
<ref="subject"/>  
<ref="place"/>  
<from type="From"/>
```

## Definitions

### **startDate**

Specifies the start of the block.

### **endDate**

Specifies the end of the block.

### **acceptLevel**

Specifies the acceptLevel for the block of time. For example, is the user free, busy, etc.

### **subject**

Specifies the subject. The subject is available only if the user doing the Busy Search has proxy rights to the recipient in the Busy Search list.

### **place**

Specifies the place in the FreeBusyBlock.

### **from**

Specifies from recipient in the FreeBusyBlock.



# FreeBusyBlockList

Lists free/busy blocks of time.

## Syntax

```
<FreeBusyBlockList>  
<block type="FreeBusyBlock" minOccurs="0" maxOccurs="unbounded"/>
```

# FreeBusyInfo

Describes the free/busy time for a user as a list of [FreeBusyBlock](#) elements.

## Syntax

```
<FreeBusyInfo>
  <extension base="NameAndEmail"/>
  <recipType type="RecipientType"/>
  <blocks type="FreeBusyBlockList"/>
</FreeBusyInfo>
```

## Definitions

### NameAndEmail

FreeBusyInfo extends [NameAndEmail](#).

### recipType

Specifies the type of recipient: In the case of FreeBusy, the type can only be User or Resource.

**--For GroupWise2012 and later.**

### blocks

Specifies a list of free/busy blocks.

# FreeBusyInfoList

Contains a list of [FreeBusyInfo](#) elements.

## Syntax

```
<FreeBusyInfoList>  
<user type="FreeBusyInfo" minOccurs="0" maxOccurs="unbounded"/>
```

# FreeBusyStats

Contains the total number of users in the Busy Search. It also specifies the number of users that have responded and those that are outstanding.

## Syntax

```
<FreeBusyStats>  
<responded" type="int"/>  
<outstanding" type="int"/>  
<total" type="int"/>
```

## Definitions

### **responded**

Specifies the total number of users that have responded with free/busy information.

### **outstanding**

Specifies the total number of users that have not responded with free/busy information.

### **total**

Specifies the total number of users in the Busy Search.

# FreeBusyUserList

Contains a list of FreeBusyUser elements.

## Syntax

```
<FreeBusyUserList>  
<user type="FreeBusyUser" minOccurs="0" maxOccurs="unbounded"/>
```

# Frequency

Contains an enumeration of the frequency of a recurrence rule.

## Syntax

```
<Frequency>  
<enumeration value="Daily"/>  
<enumeration value="Weekly"/>  
<enumeration value="Monthly"/>  
<enumeration value="Yearly"/>
```

## Definitions

### Daily

Specifies the frequency as daily.

### Weekly

Specifies the frequency as weekly.

### Monthly

Specifies the frequency as monthly.

### Yearly

Specifies the frequency as yearly.

# From

Identifies the user that sent an item.

## Syntax

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

## Definitions

### NameAndEmail

From extends [NameAndEmail](#).

### replyTo

Specifies the IMAP replyTo field.

# FullName

Specifies a list of the various parts of a user's name.

## Syntax

```
<FullName>  
<ref="displayName" />  
<namePrefix type="string" />  
<firstName type="string" />  
<middleName type="string" />  
<lastName type="string" />  
<nameSuffix type="string" />
```

## Definitions

### **displayName**

Specifies the displayable name, which is a possible combination of the other name parts.

### **namePrefix**

Specifies the prefix (Mr., Miss, Mrs., etc.).

### **firstName**

Specifies the first name.

### **middleName**

Specifies the middle name (or initials).

### **lastName**

Specifies the surname or last name.

### **nameSuffix**

Specifies the suffix (Jr., III, etc.).



# GeneralAccount

The general account settings. --For GroupWise 2012 and later.

## Syntax

```
<GeneralAccount>
  <extension base="Item">
    <ref="displayName"/>
    <warnAttachmentLimit_KB type="int"/>
    <warnExternalHTMLImages type="HTMLImages"/>
    <online type="PollSettings"/>
    <remote type="PollSettings"/>
    <cache type="PollSettings"/>
    <NNTP type="NNTPSettings"/>
  </GeneralAccount>
```

## Definitions

### Item

Inherits from the [Item](#) element.

### displayName

Specifies the display name for the account.

### warnAttachmentLimit\_KB

Specifies the attachment size in kilobytes before warning.

### warnExternalHTMLImages

Setting for warning about external HTML images.

### online

Specifies Online mode poll settings.

### remote

Specifies Remote mode poll settings.

### cache

Specifies Caching mode poll settings.

### NNTP

Specifies NNTP settings.

## GMTOffset

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

### Syntax

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

# Group

Identifies address book groups.

## Syntax

```
<Group>  
<extension base="AddressBookItem">  
<members type="GroupMemberList"/>  
<ref="email"/>
```

## Definitions

### AddressBookItem

Group extends [AddressBookItem](#).

### members

Specifies the list of group members.

### email

Specifies the email address.

# GroupMember

Identifies a group member.

## Syntax

```
<GroupMember>
  <ref="id"/>
  <ref="sid"/>
  <ref="name"/>
  <ref="e-mail"/>
  <distType type="DistributionType" default="TO"/>
  <itemType type="ContactType" default="Contact"/>
  <members type="GroupMemberList"/>
</GroupMember>
```

## Definitions

### id

Specifies the unique identifier for the member.

### sid

Specifies a short unique identifier for the member.

**--For GroupWise 8.0 HP1 and later.**

### name

Specifies the displayable name.

### e-mail

Specifies the email address of the member.

### distType

Specifies the addressing line this member belongs to (TO, BC, or CC).

### itemType

Specifies the address book item type.

### members

Specifies a group within a group.

# GroupMemberList

Contains a list of group members.

## Syntax

```
<GroupMemberList>  
<member type="GroupMember" minOccurs="0" maxOccurs="unbounded"/>
```

# Header

Specifies the column header display settings.

## Syntax

```
<Header>  
<field type="string"/>  
<width type="int"/>
```

## Definitions

### field

Specifies the name of the field.

### width

Specifies the width of the field in pixels.

# Host

Specifies an IP address or DNS name and port of a remote computer.

## Syntax

```
<Host>  
<ipAddress type="string"/>  
<port type="unsignedInt"/>
```

## Definitions

### ipAddress

Specifies the IP address of a remote computer.

### Port

Specifies the port number of a remote computer.

# Hour

Restricts to an hour value to be between 0 and 23.

## Syntax

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



# HTMLImages

Specifies values used in warning of external HTML images.

## Syntax

```
<HTMLImages>
  <enumeration value="always"/>
  <enumeration value="never"/>
  <enumeration value="checkFrequentContacts"/>
</HTMLImages>
```

## Definitions

### service

Specifies the service string (ICQ, MSN, etc.).

### name

Specifies the name of the account.

### type

Specifies the type of instant messaging account: work, home, etc.

### primary

Specifies the primary IM address in the list.

# ImAddress

Identifies an instant messaging address.

## Syntax

```
<ImAddress>  
<service type="string"/>  
<address type="string"/>  
<type type="string"/>  
<primary type="boolean"/>
```

## Definitions

### **service**

Specifies the service string (ICQ, MSN, etc.).

### **name**

Specifies the name of the account.

### **type**

Specifies the type of instant messaging account: work, home, etc.

### **primary**

Specifies the primary IM address in the list.

# ImAddressList

Contains a list of instant message addresses.

## Syntax

```
<ImAddressList>  
<im type="ImAddress" minOccurs="0" maxOccurs="unbounded"/>
```

# IMAP

Specifies the details of an IMAP account.

## Syntax

```
<IMAP>  
<extension base="Account">  
<folder type="uid"/>  
<inbound type="Server"/>  
<outbound type="Server"/>
```

## Definitions

### folder

Specifies the folder/container id for the IMAP account.

### inbound

Specifies the inbound/incoming server, credentials and flags.

### outbound

Specifies the outbound/outgoing server, credentials and flags.

# IMAPFolder

Specifies the details of an IMAPFolder.

## Syntax

```
<IMAPFolder>  
<extension base="Folder">  
<folderType type="FolderType"/>  
<account type="IMAP"/>
```

## Definitions

### folderType

Specifies the folderType.

### account

Specifies the account information.

# Item

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

## Syntax

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

## 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.

### sid

Specifies short unique identifier for the item.

**--For GroupWise 8.0 HP1 and later.**

### 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.

# ItemChanges

Contains changes to an item. You can add, delete, or update existing fields or elements in an item.

## Syntax

```
<ItemChanges>  
<add type="Item"/>  
<delete type="Item"/>  
<update type="Item"/>
```

# ItemClass

Marks an item as Private on a send. This is the same as calling [markPrivateRequest](#) after the item exists.

## 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"/>
```

# ItemRef

Contains a reference to another object. For example, the owner of a resource.

## Syntax

```
<ItemRef>  
<attribute name="uid" type="uid"/>
```

## Definitions

### uid

Specifies the unique identifier of the reference.

# ItemRefList

Contains a list of item identifiers.

## Syntax

```
<ItemRefList>  
<item type="uid" minOccurs="1" maxOccurs="unbounded" /
```

# Items

Contains a list of items.

## Syntax

```
<Items>  
<item type="Item" minOccurs="0" maxOccurs="unbounded"/>
```

# 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 the source of an item.

## Syntax

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

## Definitions

### **received**

Specifies 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.



## ItemSourceList

Contains a space-delimited list of [ItemSource](#) enumerations. For example, "sent draft personal."

### Syntax

```
<ItemSourceList>  
<list itemType="ItemSource"/>
```

# 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.

# JunkEntry

Defines an entry in the junk handling list.

## Syntax

```
<JunkEntry>
<id type="uid"/>
<match type="string"/>
<matchType type="JunkMatchType"/>
<listType type="JunkHandlingListType"/>
<useCount type="int"/>
<lastUsed type="dateTime"/>
<element ref="version"/>
<element ref="modified"/>
```

## Definitions

### id

Specifies the ID of the junk mail entry.

### match

Specifies the email address or the domain for comparison against incoming email.

### matchType

Specifies whether the value in the match element is an email or a domain.

### listType

Specifies the type of list: junk, block, or trust.

### useCount

Specifies how many times the junkEntry has successfully moved an item into the junk, block, or trust list.

### lastUsed

Specifies the last time an item was successfully moved to a junk mail list.

### version

Specifies how many times the junkEntry has changed.

### modified

Specifies the last time the junkEntry was changed.

### JunkHandlingList

```
<JunkHandlingList>
<entry type="JunkEntry" maxOccurs="unbounded"/>
```

# JunkHandlingListType

Contains an enumeration of the different junk mail list types: junk, block, or trust.

## Syntax

```
<JunkHandlingListType>  
<enumeration value="junk"/>  
<enumeration value="block"/>  
<enumeration value="trust"/>
```

## JunkMatchType

Contains the match type: email or domain.

### Syntax

```
<JunkMatchType>  
<enumeration value="e-mail"/>  
<enumeration value="domain"/>
```

# Library

Defines a GroupWise library.

## Syntax

```
<Library>  
<extension base="Item">  
<element ref="description"/>  
<domain type="string"/>  
<postOffice type="string"/>
```

## Definitions

### Item

Extends [Item](#). Specifies the ID and name of the library.

### description

Specifies the description of the library.

### domain

Specifies the domain in which the library resides.

### postOffice

Specifies the post office in which the library resides.

# LibraryList

Contains a list of libraries.

## Syntax

```
<LibraryList>  
<library type="Library" maxOccurs="unbounded"/>
```



# LinkInfo

Is a server-defined object that is passed unchanged to required methods. Currently, the only two methods that use the linkInfo object are forward and reply.

## Syntax

```
<LinkInfo>
<id type="string"/>
<type type="LinkType"/>
<thread" type="string"/>
<copyAttachment type="boolean"/>
```

## Definitions

### id

Specifies the ID of the linked item.

### type

Specifies the type of link: forward or reply.

### thread

Specifies the discussion thread.

### copyAttachment

Specify on a reply whether to include attachments.

# LinkType

Contains an enumeration of link types.

## Syntax

```
<LinkType>
  <enumeration value="forward"/>
  <enumeration value="reply"/>
  <enumeration value="draft"/>
  <enumeration value="resend"/>
</LinkType>
```

## Definitions

### forward

Specifies the forward link type. This LinkType is created on the server when the [forwardRequest](#) method is called.

### reply

Specifies the reply link type. This LinkType is created on the server when the [replyRequest](#) method is called.

### draft

Specifies the draft link type. This LinkType is used to specify using an existing draft message as the template for a item to send.

**--For GroupWise 8.0 SP2 and later.**

### resend

Specifies the resend link type. This LinkType is created on the server when the [sendItemRequest](#) method is called.

**--For GroupWise 2012 and later.**

# Mail

Contains a base mail message object. Include the "internet" keyword in the view to retrieve information indicating if the source of the item is from the Internet.

## 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"/>
    <junkMailEvaluation type="TextLines"/>
    <junkMailSettings type="TextLines"/>
    <globalSignatureInternalId type="string"/>
    <totals type="SendTotals"/>
    <attribute name="internet" type="boolean"/>
    <attribute name="stub" type="boolean"/>
    <attribute name="mime" type="boolean"/>
  </Mail>
```

## Definitions

### subject

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

### 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 the following: "name=value" (for example, "xfieldName=xFieldValue"). The xField is stored in the item. The xField are sent out via 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**

The retentionModified timestamp value is stored in the mailbox. The value is returned and set with getTimestampRequest and setTimestampRequest.

Each item has a retentionModified value. The item value is compared against the mailbox retentionModified value. If the item value is less than the mailbox value, the item can be purged. That means the item has been retained. That is, of course, if retention is enabled.

The item retentionModified value is updated whenever the subject, personal subject or attachment values are changed.

**rssURL**

Specifies the RSS URL of the RSS item.

**junkMailEvaluations**

Specifies how the item was evaluated for being junk mail.

**--For GroupWise 2012 and later.**

**junkMailSettings**

Specifies the junk mail settings.

**--For GroupWise 2012 and later.**

**globalSignatureInternalId**

Specifies the internal id of the global signature.

**totals**

Specifies the totals for sent items.

**internet**

Specifies that the item came in through the internet (GWIA).

**stub**

Specifies if the item is a stubbed/archived item.

**mime**

Specifies that the item has a mime RFC822 attachment to use in formatting the item to send.

**--For GroupWise 8.0 SP1 and later.**

# 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"/>
    <attribute name="path" type="string"/>
  </extension>
</MessagePart>
```

## 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

For more information on hash values see [AttachmentFlags](#).

### path

For more information on path see [AttachmentFlags](#).

**--For GroupWise 8.0 SP1 and later.**

# MessageType

Contains an enumeration of item types.

## Syntax

```
<MessageType>
<enumeration value="Appointment"/>
<enumeration value="CalendarItem"/>
<enumeration value="DocumentReference"/>
<enumeration value="Mail"/>
<enumeration value="Note"/>
<enumeration value="PhoneMessage"/>
<enumeration value="Task"/>
```

## Definitions

### Appointment

Specifies that the item is an appointment.

### CalendarItem

Specifies that the item is an appointment, note, or task.

### DocumentReference

Specifies that the item is a document reference.

### Mail

Specifies that the item is a mail item.

### Note

Specifies that the item is a note.

### PhoneMessage

Specifies that the item is a phone item.

### Task

Specifies that the item is a task.



# MessageTypeList

Contains a space-delimited list of [MessageType](#) elements. For example, Appointment, PhoneMessage, or Note.

## Syntax

```
<MessageTypeList>  
<list itemType="MessageType" />
```

# Minute

Restricts a minute to be between 0 and 59.

## Syntax

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

# ModifyItem

Modifies an existing item.

## Syntax

```
<ModifyItem>  
<id type="uid"/>  
<notification type="SharedFolderNotification" minOccurs="0"/>  
<updates type="ItemChanges"/>
```

## Definitions

### id

Specifies the ID of the item to modify.

### notification

Specifies to modify a shared folder notification.

### updates

Specifies the updates to make to the item.

# Month

Restricts months to be between 0 and 11.

## Syntax

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

# MonthList

Specifies a list of [Month](#) elements.

## Syntax

```
<MonthList>  
<month type="Month" minOccurs="0" maxOccurs="unbounded"/>
```

# MoveItem

Provides a structure to reference a single item in the [moveItemsRequest](#). --For GroupWise 2012 and later.

## Syntax

```
<MoveItem>
  <id type="uid">
  <container type="uid"/>
  <from type="uid"/>
</MoveItem>
```

## Definitions

### id

Specifies a unique identifier for the item to move or link.

### container

Specifies a unique identifier for the folder to put the item.

### from

(optional) Specifies a unique identifier for the folder the item is moved from. If not specified, the item is copied (linked) to the new folder.

# 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.

# Nickname

Specifies a GroupWise Nickname. A Nickname is an alternative email for an existing GroupWise object.

## Syntax

```
<Nickname>  
<extension base="AddressBookItem">  
<ref="email"/>  
<realType type="ContactType"/>  
<owner type="Owner"/>
```

## Definitions

### email

Specifies the email address for the Nickname.

### realType

Specifies the item type of the Nickname (for example, Domain, Post Office, User).

### owner

Specifies the owner/visibility of the Nickname (for example, Domain, Post Office, User).



# NNTP

Specifies the details of an NNTP account.

## Syntax

```
<NNTP>  
<extension base="Account">  
<folder type="uid"/>  
<inbound type="Server"/>  
<outbound type="Server"/>  
<life type="int"/>  
<maxMessages type="int"/>  
<newMessagesXDaysOld type="int"/>
```

## Definitions

### folder

Specifies the folder/container id for the NNTP account.

### inbound

Specifies the inbound/incoming server, credentials and flags.

### outbound

Specifies the outbound/outgoing server, credentials and flags.

### life

Specifies the life of a message in days. Remove messages after “life” days.

### maxMessages

Specifies the maximum number of messages to be downloaded at one time.

### newMessagesXDaysOld

Download message X days old.

# NNTPFolder

Specifies the NNTP folder.

## Syntax

```
<NNTPFolder>  
<extension base="Folder">  
<folderType type="FolderType"/>  
<account type="NNTP"/>
```

## Definitions

### **folderType**

Specifies the folderType.

### **account**

Specifies the account information.

# NNTPSettings

Specifies NNTP settings. --For GroupWise 2012 and later.

## Syntax

```
<NNTPSettings>
  <usePlainText type="boolean">
  <maxLine type="int"/>
  <indentChar type="int"/>
  <category type="uid"/>
</NNTPSettings>
```

## Definitions

### usePlainText

Specifies whether or not to use plain text for the message body.

### maxLine

Specifies the number of characters in a line of the message body before folding the line.

### indentChar

Specifies the character to use in indenting the message body on a reply.

### category

Specifies the category to mark on an NNTP item.

## Note

Contains a GroupWise reminder note. Note extends [CalendarItem](#).

### Syntax

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

### Definitions

#### **startDate**

Specifies the starting date for the note.

# NotificationType

Contains an enumeration of the types of notifications.

## Syntax

```
<NotificationType>  
  <enumeration value="SharedAddressBook"/>  
  <enumeration value="SharedFolder"/>  
  <enumeration value="SharedCalendar"/>  
</NotificationType>
```

## Definitions

### **sharedAddressBook**

Specifies a shared address book notification.

### **SharedFolder**

Specifies a shared folder notification.

### **SharedCalendar**

Specifies a shared calendar notification.

# NotifyEntry

Contains a structure to define a user to receive notifications. --For GroupWise 2012 and later.

## Syntax

```
<NotifyEntry>  
  <extension base="NameAndEmail">  
    <ref="id"/>  
    <rights type="Subscribe"/>  
</NotifyEntry>
```

## Definitions

### id

Specifies a unique identifier for the structure.

### rights

Specifies subscription rights.

# NotifyEntryChanges

Contains the structure used in changing the notification rights. --For GroupWise 2012 and later.

## Syntax

```
<ItemChanges>
  <add type="NotifyEntry"/>
  <delete type="NotifyEntry"/>
  <update type="NotifyEntry"/>
</ItemChanges>
```

## Definitions

### add

Specifies the elements to add.

### delete

Specifies the elements to delete.

### update

Specifies the elements to update.

# NotifyList

Contains a list of [NotifyEntry](#) elements.

## Syntax

```
<NotifyList>  
  <entry type="NotifyEntry" maxOccurs="unbounded" />  
</NotifyList>
```

## Definitions

### entry

Specifies the NotifyEntry definition.



# 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"/>
```

# OfficeInfo

Contains miscellaneous office information for a contact.

## Syntax

```
<OfficeInfo>
<organization type="ItemRef"/>
<department type="string"/>
<title type="string"/>
<website type="anyURI"/>
<profession type="string"/>
<location type="string"/>
<manager type="string"/>
<assistant type="string"/>
```

## Definitions

### organization

Specifies the reference to an organization (Company).

### department

Specifies the department name.

### title

Specifies the title string.

### website

Specifies the Web site string.

### profession

Specifies a contact's profession.

### location

The location element maps to the Advanced Tab > Location field on a GW Contact.

---

**NOTE:** In GroupWise 8.0, the location element in the postal address is confusing. On a contact in the GroupWise 7.0 Windows client, the location element mapped to the Mail Stop on the office address. In GroupWise 8.0 Windows client, it maps to the Details > Office > Location on a Contact. The mapping issue was not corrected because it could disable existing third party solutions.

---

### manager

Specifies a contact's manager.

### assistant

Specifies a contact's assistant.

# Organization

Contains contact information for a company.

## Syntax

```
<Organization>
<extension base="AddressBookItem"
<contact type="ItemRef"/>
<phone type="string"/>
<fax type="string"/>
<address type="PostalAddress"/>
<website type="anyURI"/>
```

## Definitions

### **contact**

Specifies the contact person for the organization.

### **phone**

Specifies the phone number for the organization.

### **fax**

Specifies the fax number for the organization.

### **address**

Specifies the mailing address for the organization.

### **website**

Specifies the organization's Web site.

# Owner

Specifies the Owner of an object.

## Syntax

```
<Owner>  
<domain type="string"/>  
<postOffice type="string"/>  
<userid type="string"/>
```

## Definitions

### **domain**

Specifies domain visibility.

### **postOffice**

Specifies post office visibility.

### **userid**

Specifies a user id.

# PanelDisplaySettings

Specifies the display settings for a panel. See the section [DisplaySettings](#) on how to retrieve the panel settings.

## Syntax

```
<PanelDisplaySettings>
<extension base="DisplaySettings">
<panelName type="string"/>
<column type="int"/>
<index type="int"/>
<container type="uid"/>
<filter type="Filter"/>
<splitBar type="string"/>
<width type="int"/>
<height type="int"/>
<originalHeight type="int"/>
<weekViewColumns type="int"/>
<calendarView type="CalendarViewType"/>
<website type="string"/>
<interval type="unsignedInt"/>
<hScrollPosition type="int"/>
<vScrollPosition type="int"/>
<element ref="description"/>
```

## Definitions

### panelName

Specifies the name of the panel.

### column

Specifies the column.

### index

Specifies the index.

### container

Specifies the container/folder.

### filter

Specifies the filter.

### splitBar

Specifies the splitBar.

### width

Specifies the width.

### height

Specifies the height.

**originalHeight**

Specifies the originalHeight.

**weekViewColumns**

Specifies the weekViewColumns.

**calendarView**

Specifies the calendar view.

**website**

Specifies the website.

**interval**

Specifies the update interval.

**hScrollPosition**

Specifies the hScrollPosition.

**vScrollPosition**

Specifies the vScrollPosition.

**description**

Specifies the description.

# PanelList

Specifies a list of panels.

## Syntax

```
<PanelList>  
<panel type="PanelDisplaySettings"/>
```

# PersonalInfo

Contains miscellaneous personal information for a contact.

## Syntax

In the PersonalInfo section, you can specify a birthday or anniversary date. There is a problem if the client specifying the date is in a different time zone than the POA. The timezone element allows you to specify the timezone to use. Thus, the birthday and anniversary events will show on the correct day.

```
<PersonalInfo>
  <birthday type="date"/>
  <website type="string"/>
  <spouse type="string"/>
  <children type="string"/>
  <hobbies type="string"/>
  <nickName type="string"/>
  <picture type="PictureData"/>
  <weddingAnniversary type="date"/>
  <displayBirthdayYear type="boolean"/>
  <displayAnniversaryYear type="boolean"/>
  <timezone type="tns:Timezone"/>
</PersonalInfo>
```

## Definition

### spouse

Specifies the spouse of the contact.

### children

Specifies the children of the contact.

### hobbies

Specifies the hobbies of the contact.

### nickName

Specifies the nickName of the contact.

### picture

Specifies a picture of a contact. See [PictureData](#) for more information.

### weddingAnniversary

Specifies the weddingAnniversary of the contact.

### displayBirthdayYear

Specifies if the birthday year should be displayed.

--For GroupWise 2012 and later.



**displayAnniversaryYear**

Specifies if the anniversary year should be displayed.

**--For GroupWise 2012 and later.**

**timezone**

Specifies the timezone to use in the creation of the birthday or anniversary events. This is the same value used in the creation of Appointments.

**--For GroupWise 2012 and later.**

# 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"/>
```

## PhoneList

Contains a list of phone numbers for a contact. The default number must exist in the list.

### Syntax

```
<PhoneList>  
<phone type="PhoneNumber" minOccurs="0" maxOccurs="unbounded"/>  
<attribute name="default"/>
```

# PhoneMessage

Represents a GroupWise phone message.

## Syntax

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

# PhoneNumber

Contains the phone number for a contact.

## Syntax

```
<PhoneNumber>  
<attribute name="type" type="PhoneNumberType" use="required"/>
```

## Definitions

### type

Specifies the type of phone number.

# PhoneNumberType

Contains an enumeration of the types of phone numbers.

## Syntax

```
<PhoneNumberType>
<enumeration value="Fax" />
<enumeration value="Home" />
<enumeration value="Mobile" />
<enumeration value="Office" />
<enumeration value="Pager" />
<enumeration value="CallBack" />
<enumeration value="Office2" />
<enumeration value="Radio" />
<enumeration value="Car" />
<enumeration value="Other" />
<enumeration value="Telex" />
<enumeration value="ISDN" />
<enumeration value="Assistant" />
<enumeration value="Home2" />
<enumeration value="FaxOffice" />
<enumeration value="FaxHome" />
<enumeration value="TTYTDD" />
<enumeration value="Company" />
```

# PictureData

Specifies the content of a picture.

## Syntax

```
<PictureData>  
<contentType type="xs:string"/>  
<size type="xs:unsignedInt"/>  
<data type="xs:base64Binary"/>
```

## Definitions

### contentType

Specifies the contentType of the image. It should be "image/jpeg".

### size

Specifies the size of the image. The image needs to be smaller than 2K.

### data

Specifies the actual image data.

## Remarks

A picture has the following restrictions. It is up to the developer to make sure the image meets the following criteria.

- ♦ The file format is jpeg. The contentType specifies the mime contentType of "image/jpeg".
- ♦ The size of the file must be less than 2K.
- ♦ The maximum dimension of the image is 64 X 64 pixels.

# PlainText

Contains the plain text authorization. The PlainText element is used to provide a user name and password to log into GroupWise. **--WARNING: The password is sent in clear text. If you want to protect the passwords, you need to configure the POA to use HTTPS.**

## Syntax

```
<PlainText>  
<extension base="Authentication">  
<password" type="string"/>
```

## Definitions

### password

Specifies the GroupWise password.



# PollSettings

Contains settings for how and when the GroupWise client should query the POA. For example, how often (interval) should the GroupWise client in Caching mode sync to the POA. **--For GroupWise 2012 and later.**

## Syntax

```
<PollSettings>
  <pollAtStartup type="boolean"/>
  <pollMarkedAccounts type="boolean"/>
  <pollInterval type="int"/>
</PollSettings>
```

## Definitions

### **pollAtStartup**

Specifies whether or not to poll the POA at startup.

### **pollMarkedAccount**

Specifies whether or not to poll the marked accounts.

### **pollInterval**

Specifies how often to poll the POA.

# PostalAddress

Contains a mailing address.

## Syntax

```
<PostalAddress>
<ref="description"/>
<streetAddress type="string"/>
<location type="string"/>
<city type="string"/>
<state type="string"/>
<postalCode type="string"/>
<country type="string"/>
<attribute name="type" type="PostalAddressType"/>
<poBox type="string"/>
```

## Definitions

### location

In GroupWise 8.0, the location element in the postal address is confusing. On a contact in the GroupWise 7.0 Windows client, the location element mapped to the Mail Stop on the office address. In GroupWise 8.0 Windows client, it now maps to the Details > Office > Location on a Contact. The mapping issue was not corrected because it could disable existing third party solutions.

### poBox

Specifies the poBox.

# PostalAddressList

Contains a list of addresses.

## Syntax

```
<PostalAddressList>  
  <address type="PostalAddress" minOccurs="0" maxOccurs="unbounded"/>  
  <attribute name="mailingAddress" type="PostalAddressType"/>  
</PostalAddressList>
```

## Definitions

### PostalAddress

Specifies 0 or more [PostalAddress](#) elements.

### mailingAddress

Specifies the postal address type (for example, Home, Office, Other).

# PostalAddressType

Contains an enumeration of the types of a postal address.

## Syntax

```
<PostalAddressType>  
<enumeration value="Home"/>  
<enumeration value="Office"/>  
<enumeration value="Other"/>
```

## Definitions

### other

For GroupWise 8.0 and later.

# ProblemEntry

Reports errors on individual items.

## Syntax

```
<ProblemEntry>  
<element type="string"/>  
<ref="code"/>  
<ref="description"/>
```

# ProblemList

Contains a list of [ProblemEntry](#) elements. If a request specifies more than one item, ProblemList returns error information for the specific items.

## Syntax

```
<ProblemList>  
<entry" type="ProblemEntry" minOccurs="0" maxOccurs="unbounded" />
```

# Proxy

Contains the proxy authentication.

## Syntax

```
<Proxy>  
<extension base="Authentication"  
<password type="string"/>  
<proxy type="string"/>  
<mainVersion type="xs:string" minOccurs="0"/>
```

## Definitions

### password

Specifies the proxying user's password.

### proxy

Specifies the user's ID that granted the access rights.

### mainVersion

Specifies the version of the proxying user's [loginRequest](#). This ensures that the schema versions match between the proxying user's mailbox and the mailbox that the proxying user logs into.

**--For GroupWise 2012 SP1 and later.**

# ProxyFolder

Represents a proxy user's Calendar folder. --For GroupWise 2012 and later.

## Syntax

```
<ProxyFolder>
  <extension base="Folder">
    <folderType type="FolderType"/>
    <proxy type="NameAndEmail"/>
  </ProxyFolder>
```

## Definitions

### folderType

Identifies the folder type.

### proxy

Identifies the proxied user.



# ProxyList

Contains a list of users that you have proxied to in the past.

## Syntax

```
<ProxyList>  
<proxy type="NameAndEmail" minOccurs="0" maxOccurs="unbounded"/>
```

## Definitions

### **proxy**

Specifies the GroupWise login name of a user that granted the proxy rights and that you have also successfully proxied.

# Query

Contains a query definition.

## Syntax

```
<Query>  
<element name="target" type="QueryTargetList"/>  
<filter type="Filter"/>
```

## Definitions

### **target**

Specifies the source to search.

### **filter**

Specifies the filter.

# QueryFolder

Contains a query folder.

## Syntax

```
<QueryFolder>  
<extension base="Folder">  
<folderType type="FolderType"/>  
<query type="Query"/>
```

# QueryTarget

Contains a query target.

## Syntax

```
<QueryTarget>  
<source type="SourceType" />  
<info type="UserInfo" />  
<container type="uid" />
```

## Definitions

### container

Specifies the folder to search.

### source

Specifies the source (Library or User).

### info

Specifies the user information.

# QueryTargetList

Specifies a list of query targets.

## Syntax

```
<QueryTargetList>  
<target type="QueryTarget" />
```

## Definitions

### target

Specifies a list of query targets.

# RangeDays

Contains the days to ask for from the calendar host.

## Syntax

```
<RangeDays>  
  <previous type="unsignedInt"/>  
  <through type="unsignedInt"/>  
</RangeDays>
```

## Definitions

### previous

Specifies days before today.

### through

Specifies days after today.

# 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"/>
  <domain type="string"/>
  <postOffice type="string"/>
  <idomain type="string"/>
</Recipient>
```

## 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 that have opened and accepted appointments.

### **acceptLevel**

Specifies the acceptLevel for free/busy time.

### **domain**

Specifies the domain of the recipient. Must log in with <version>1.05</version> or later.

**--For GroupWise 2012 and later.**

### **postOffice**

Specifies the post office of the recipient. Must log in with <version>1.05</version> or later.

**--For GroupWise 2012 and later.**

### **idomain**

Specifies the Internet domain of the recipient. Must log in with <version>1.05</version> or later.

**--For GroupWise 2012 and later.**

# RecipientList

Contains a list of [Recipient](#) elements.

## Syntax

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



# RecipientStatus

Contains status information on items that were sent to other users. For example, the sender can track users that have opened, read, and accepted their appointment

## Syntax

```
<RecipientStatus>
<delivered type="dateTime"/>
<undeliverable type="dateTime"/>
<transferred type="dateTime"/>
<transferDelayed type="dateTime"/>
<transferFailed type="TransferFailedStatus"/>
<downloaded type="dateTime"/>
<downloadedByThirdParty type="dateTime"/>
<retractRequested type="dateTime"/>
<retracted type="dateTime"/>
<opened type="dateTime"/>
<deleted type="dateTime"/>
<undeleted type="dateTime"/>
<purged type="dateTime"/>
<accepted type="CommentStatus"/>
<declined type="CommentStatus"/>
<replied type="dateTime"/>
<forwarded type="dateTime"/>
<shared type="dateTime"/>
<started type="dateTime"/>
<completed type="dateTime"/>
<incomplete type="dateTime"/>
<delegated type="DelegatedStatus"/>
<delegateeStatus type="DelegateeStatus" maxOccurs="unbounded"/>
```

## Definitions

### **delivered**

Specifies the delivery date.

### **undeliverable**

Specifies that the item was undeliverable.

### **transferred**

Specifies that the item was successfully transferred outside the GroupWise system.

### **transferDelayed**

Specifies that there is a delay in transferring the item outside the GroupWise system.

### **transferFailed**

Specifies the reason why the transfer outside the GroupWise system failed.

### **downloaded**

Specifies that the item has been downloaded by a GroupWise client to a remote or caching mailbox.

**downloadedByThirdParty**

Specifies that the item has been downloaded by a third-party application.

**retractRequested**

Specifies that a retraction of the item from the recipient's mailbox is in progress.

**retracted**

Specifies that the retraction of the item from the recipient's mailbox has been successful.

**opened**

Specifies that the recipient opened the item.

**deleted**

Specifies that the recipient deleted the item.

**undeleted**

Specifies that the recipient undeleted the item.

**purged**

Specifies that the recipient purged the item.

**accepted**

Specifies that the recipient accepted the item. There is an optional comment field sent from the recipient during the accept action.

**declined**

Specifies that the recipient declined the item. There is an optional comment field sent from the recipient during the decline action.

**replied**

Specifies that the recipient replied the item.

**forwarded**

Specifies that the recipient forwarded the item.

**shared**

Specifies that the recipient shared the item.

**started**

Specifies that the recipient started a workflow item.

**completed**

Specifies that the recipient completed the item.

**incomplete**

Specifies that the recipient has not completed the item.

**delegated**

Specifies that the recipient delegated the item.

## Remarks

When you send out a calendar item, recipients can delegate the item. As the sender, you can see the status of the original recipients as well as the recipient the original item was delegated to (delegatee) with the following:

```
<delegateeStatus type="DelegateeStatus" 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"/>
```

# RecurrenceDateType

Contains a list of recurrence dates.

## Syntax

```
<RecurrenceDateType>  
<date type="date" maxOccurs="unbounded"/>
```

# RecurrenceRule

Follows the same logic as iCalendar (RFC 2445).

## Syntax

```
<RecurrenceRule>
  <frequency type="Frequency"/>
  <count type="unsignedInt"/>
  <until type="date"/>
  <interval type="unsignedInt"/>
  <byDay type="DayOfWeekList"/>
  <byMonthDay type="DayOfMonthList"/>
  <byYearDay type="DayOfYearList"/>
  <byMonth type="MonthList"/>
  <bySetPos type="DayOfYearList"/>
  <honorRule type="boolean"/>
</RecurrenceRule>
```

## Definitions

### frequency

Specifies the frequency of the recurrence rule.

### count

Specifies the number of items that were created with the recurrence rule.

### until

Specifies the ending date of the recurrence rule.

### interval

Specifies the interval between created items, based on the frequency.

### byDay

Specifies a list of days.

### byMonthDay

Specifies a list of the days of the month.

### byYearDay

Specifies a list of the days of the year.

### byMonth

Specifies a list of the months of the year.

### bySetPos

Specifies whether to set the date information by position.

--For GroupWise 2012 and later.

**honorRule**

By default, the startDate is always put in the recurrence set, but if honorRule is true, the startDate value is only put in if it matches the rule.

**--For GroupWise 2012 and later.**

## ReferenceInfo

Contains the date the item was last referenced and the number of times the item has been referenced are tracked for the Frequent Contacts address book.

### Syntax

```
<ReferenceInfo>  
<lastReferenceDate" type="dateTime" />  
<referenceCount" />
```

### Definitions

#### **lastReferenceDate**

Specifies the date the item in the Frequent Contacts book was last referenced.

#### **referenceCount**

Specifies the number of times the item in the Frequent Contacts book has been referenced.



# Resource

Identifies a GroupWise resource in an address book.

## Syntax

```
<Resource>
<extension base="AddressBookItem"
<phone type="string"/>
<resourceType type="string"/>
<ref="e-mail"/>
<owner type="ItemRef"/>
```

## Definitions

### phone

Specifies the phone number.

### resourceType

Specifies the type of the resource.

### e-mail

Specifies the email address of the resource.

### owner

Specifies the owner of the resource.

# RetractType

Contains an enumeration of the retractType options.

## Syntax

```
<RetractType>  
<enumeration value="myMailbox" />  
<enumeration value="recipientMailboxes" />  
<enumeration value="allMailboxes" />
```

## Definitions

### **myMailbox**

Specifies to retract the item only from my mailbox and to leave the item in the recipients' mailboxes.

### **recipientMailboxes**

Specifies to retract the item only from recipients' mailboxes and to leave the item in the sender's mailboxes.

### **allMailboxes**

Specifies to retract the item from all mailboxes.

# ReturnNotification

Sets the status tracking information to receive for messages that are sent to others.

## Syntax

```
<ReturnNotification>  
<opened type="ReturnNotificationOptions"/>  
<deleted type="ReturnNotificationOptions"/>  
<accepted type="ReturnNotificationOptions"/>  
<declined type="ReturnNotificationOptions"/>  
<completed type="ReturnNotificationOptions"/>
```

## Definitions

### **opened**

Specifies that the item was opened by the recipient (available for all box entry item types)

### **deleted**

Specifies that the item was deleted by the recipient (available for all box entry item types).

### **accepted**

Specifies that the item was opened by the recipient (available for all calendar item types).

### **declined**

Specifies that the item was opened by the recipient (available for all calendar item types).

### **completed**

Specifies that the item was opened by the recipient (available for task item types).

# ReturnNotificationOptions

Defines the type of notification to return to the sender when returnNotification status tracking is enabled.

## Syntax

```
<ReturnNotificationOptions>  
<mail type="boolean"/>  
<notify type="boolean"/>
```

## Definitions

### mail

Specifies to send a mail message to the sender when the event occurs.

### notify

Specifies to notify the sender in GroupWise Notify when the event occurs.

# Rights

Contains the rights to an object.

## Syntax

```
<Rights>
<read type="boolean"/>
<add type="boolean"/>
<edit type="boolean"/>
<delete type="boolean"/>
<share type="boolean"/>
<manage type="boolean"/>
```

# Rule

Contains a GroupWise rule. Rules act on items before they are added to a mailbox.

## Syntax

```
<Rule>
<extension base="tns:ContainerItem">
<execution type="Execution"/>
<element ref="sequence"/>
<enabled type="boolean" default="0"/>
<types type="xs:string"/>
<source type="xs:string"/>
<conflict type="AppointmentConflict" default="Ignore"/>
<filter type="Filter"/>
<actions type="RuleActionList"/>
```

## Definitions

### Execution

Specifies the execution time of the rule. For example, execute on startup of the client.

### sequence

Specifies the position of the rule in the rule list.

### enabled

Specifies to enable or disable the rule. The rule is disabled by default.

### types

Specifies which item types the rule acts on. For example, the rule acts on mail and task items.

### source

Specifies which source types the rule acts on. For example, the rule acts on received and draft items.

### conflict

Specifies how to handle appointment conflicts when processing the rule.

### filter

Specifies a filter.

### actions

Specifies the action to perform.

### default

Specifies that the rule is the default action.

# RuleAction

Contains the action that should be performed when the rule condition is True.

## Syntax

```
<RuleAction>
<type type="RuleActionType"/>
<container type="uid"/>
<item type="Mail"/>
<message type="string"/>
<element ref="acceptLevel"/>
<categories type="CategoryRefList"/>
```

## Definitions

### type

Specifies what rule action type to execute.

### container

Specifies the container on a move or a link to a folder rule action type.

### item

Specifies the item (with all its elements) to send when a rule condition is true. An item is needed when the following rule actions types are specified: send, delegate, forward, reply, and send.

### message

Specifies a comment for the following rule actions types: accept, delete, decline, and delegate.

### acceptLevel

Specifies the acceptLevel for the accept rule action type.

### categories

Specifies categories for the category rule action type.

# RuleActionList

Contains a list of rule actions.

## Syntax

```
<RuleActionList>  
<action type="RuleAction" maxOccurs="unbounded"/>
```



# RuleActionType

Contains the action type that should be performed when the rule condition is True.

## Syntax

```
<RuleActionType>
  <enumeration value="Accept" />
  <enumeration value="Archive" />
  <enumeration value="Category" />
  <enumeration value="Delegate" />
  <enumeration value="Delete" />
  <enumeration value="Forward" />
  <enumeration value="Link" />
  <enumeration value="MarkPrivate" />
  <enumeration value="MarkRead" />
  <enumeration value="MarkUnread" />
  <enumeration value="Move" />
  <enumeration value="Purge" />
  <enumeration value="Reply" />
  <enumeration value="Send" />
  <enumeration value="StopRules" />
  <enumeration value="FlatForward" />
  <enumeration value="SimpleForward" />
  <enumeration value="Unknown" />
</RuleActionType>
```

# RuleList

Contains a list of rules.

## Syntax

```
<RuleList>  
<rule type="Rule" maxOccurs="unbounded"/>
```

# SendOptions

Contains the miscellaneous GroupWise send options.

## Syntax

```
<SendOptions>
  <requestReply type="SendOptionsRequestReply"/>
  <mimeEncoding type="string"/>
  <statusTracking type="StatusTracking"/>
  <statusTrackingFlags type="StatusTrackingFlags"/>
  <notification type="ReturnNotification"/>
  <updateFrequentContacts type="boolean"/>
  <workFlow type="boolean"/>
  <notifyRecipients type="boolean"/>
</SendOptions>
```

## Definitions

### **requestReply**

Specifies that the sender wants a reply from the recipients.

### **mimeEncoding**

Specifies the mimeEncoding for the item.

### **statusTracking**

Specifies whether the GroupWise system automatically creates a sent item to track the recipients' actions on the item and collects the status information for all GroupWise recipients. For example, if the item was delivered and opened by the recipients.

### **statusTrackingFlags**

We needed refine the statusTracking values. This option allows us to set the particular bits.

**--For GroupWise 2012 and later.**

### **notification**

Specifies the returnNotification options.

### **updateFrequentContacts**

Specifies to update the Frequent Contacts address book with the recipients.

### **workFlow**

Specifies if workFlow is enabled for this item. In the Windows client, workFlow is enabled on a new item during creation by enabling Actions > Routing Slip. Routing Slip and workFlow are different names for the same functionality.

### **notifyRecipients**

Specifies whether to notify recipients.

# SendOptionsRequestReply

Contains when to mark an item as Reply Requested.

## Syntax

```
<SendOptionsRequestReply>  
<whenConvenient type="boolean"/>  
<byDate type="dateTime"/>
```

## Definitions

### whenConvenient

Specifies that the sender wants a response from the recipients when it is convenient.

### byDate

Specifies that the sender wants a response from the recipients by a certain date.

# SendTotals

Specifies use counts on a sent item.

## Syntax

```
<SendTotals>
  <accepted type="unsignedInt"/>
  <alarm type="unsignedInt"/>
  <completed type="unsignedInt"/>
  <delivered type="unsignedInt"/>
  <downloaded type="unsignedInt"/>
  <hidden type="unsignedInt"/>
  <moved type="unsignedInt"/>
  <noMoreStatus type="unsignedInt"/>
  <opened type="unsignedInt"/>
  <private type="unsignedInt"/>
  <probed type="unsignedInt"/>
  <purged type="unsignedInt"/>
  <read type="unsignedInt"/>
  <replied type="unsignedInt"/>
  <retracted type="unsignedInt"/>
  <retractRequested type="unsignedInt"/>
  <started type="unsignedInt"/>
  <totalUsers type="unsignedInt"/>
  <transferred type="unsignedInt"/>
  <transferFailed type="unsignedInt"/>
  <undeliverable type="unsignedInt"/>
  <undeliverableUsers type="unsignedInt"/>
</SendTotals>
```

# Server

Specifies a server definition for an account.

## Syntax

```
<Server>  
<host type="Host"/>  
<flags type="ServerFlags"/>  
<userid type="string"/>
```

## Definitions

### host

Specifies the host configuration.

### flags

Specifies the account server flags.

### userid

Specifies a user id.

# ServerFlags

Specifies server flags for an account.

## Syntax

```
<ServerFlags>
<savePassword type="boolean"/>
<useIncomingAuth type="boolean"/>
<secureConnection type="boolean"/>
<authRequired type="boolean"/>
<IMAP_ACL type="boolean"/>
<netMail type="boolean"/>
<SSLRequired type="boolean"/>
```

## Definitions

### **savePassword**

Specifies to save the password.

### **useIncomingAuth**

Specifies to use authentication for incoming connections.

### **secureConnection**

Specifies to use a secure connection (for example, SSL).

### **authRequired**

Specifies authentication is required.

### **IMAP\_ACL**

Specifies an IMAP access control list.

### **netMail**

Specifies netMail.

### **SSLRequired**

Specifies SSL is required.

# Settings

Represents the GroupWise settings for a user. Setting are a list of names with values.

## Syntax

```
<Settings>  
<group type="SettingsGroup" maxOccurs="unbounded" />  
<setting type="Custom" maxOccurs="unbounded" />
```

## Definitions

### **group**

Specifies to group the user settings together.

### **settings**

Specifies the custom fields and their values. Administrators can lock specific settings so they cannot be changed (immutable).



# SettingsGroup

Represents a group of GroupWise settings.

## Syntax

```
<SettingsGroup>  
<setting" type="Custom" maxOccurs="unbounded" /  
<attribute name="type" type="string" /
```

# SettingsList

Represents a list of settings.

## Syntax

```
<SettingsList>  
<setting type="Custom" maxOccurs="unbounded"/>
```

# SharedBook

Specifies a shared personal address book (PAB).

## Syntax

```
<SharedBook>
<extension base="AddressBook">
  <ref="rights"/>
  <acl type="AccessControlList"/>
  <owner type="UUID"/>
  <isSharedByMe type="boolean"/>
  <isSharedToMe type="boolean"/>
</SharedBook>
```

## Definition

### rights

Specifies the rights that user2 has to a folder.

### Acl

Specifies tie rights that user1 gives to user2 during the creation of the folder.

### owner

Specifies the owner of the folder.

### isSharedByMe

Specifies if you shared this folder with other users. For example, if user1 shares a folder with user2, the isSharedByMe is True for user1.

### isSharedToMe

Specifies if another user shared this folder with you. For example, if user1 shares a folder with user2, the isSharedToMe is True for user2.

# SharedFolder

Represents a shared folder.

## Syntax

```
<SharedFolder>  
<extension base="Folder">  
<ref="rights"/>  
<acl type="FolderACL"/>  
<owner type="UUID"/>  
<isSharedByMe type="boolean"/>  
<isSharedToMe type="boolean"/>
```

## Definitions

### rights

Specifies the rights that user2 has to a folder.

### acl

Specifies the rights that user1 gives to user2 during the creation of the folder.

### owner

Specifies the owner of the folder.

### isSharedByMe

Specifies if you shared this folder with other users. For example, if user1 shares a folder with user2, the isSharedByMe is True for user1.

### isSharedToMe

Specifies if another user shared this folder with you. For example, if user1 shares a folder with user2, the isSharedToMe is True for user2.

# SharedFolderNotification

Contains the message sent to other users if the folder is shared and the rights change for the shared users. The notification can be customized by the sender.

## Syntax

```
<SharedFolderNotification>  
<ref="subject" />  
<message type="string" />  
<ref="description" />
```

## Definitions

### subject

Specifies the subject to use for the shared folder notification message.

### message

Specifies the message body text to use.

### description

Specifies the description text to use for the folder when an invitation to a shared folder is accepted.

# SharedNotification

Contains the mail message to send when a folder has been shared. The message describes the container that is being shared.

## Syntax

```
<SharedNotification>  
<extension base="Mail">  
<notification type="NotificationType"/>  
<ref="description"/>  
<ref="rights"/>
```

## Definitions

### Mail

Specifies a mail item that was created by the user that shares the folder. It is sent to the recipients of the shared folder. The notification appears in the Mailbox for the recipients. The recipients can accept or decline the shared folder notification.

### notification

Specifies the type of notification: shared folder or shared address book.

### description

Specifies a description that describes the notification.

### rights

Specifies the rights the recipients have to the shared container.

# Signature

Contains one or more HTML signatures.

## Syntax

```
<Signature>
<id type="uid"/>
<ref="name"/>
<default type="boolean" default="0"/>
<part type="SignatureData"/>
<global type="boolean" default="0"/>
```

## Definitions

### id

Specifies the unique ID of the signature.

### name

Specifies the name of the signature.

### default

Specifies if this is the default signature.

### part

Specifies the HTML signature.

### global

Is not used at this time. There is no way to retrieve the global signature via Web Services.

# SignatureData

Contains the signature's HTML text. It is Base64 encoded and a MIME document. The signatures are stored in the GroupWise store as MIME documents.

## Syntax

```
<SignatureData>  
<size type="unsignedInt" />  
<data type="base64Binary" />
```

## Definitions

### size

Specifies the size of the signature. It specifies the Base64 encoded size.

### data

Specifies the signature data in Base64. The HTML signature is a MIME document.



# Signatures

Contains a list of [Signature](#) elements.

## Syntax

```
<Signatures>  
<signature type="Signature" maxOccurs="unbounded"/>  
<setting type="SignatureSettings"/>
```

## Definitions

### setting

Specifies signature settings.

# SignatureSettings

Specifies signature settings.

## Syntax

```
<SignaturesSettings>  
<enabled type="boolean"/>  
<add type="SignatureType"/>
```

## Definitions

### enabled

Specifies whether a signature is enabled or disabled.

### add

Specifies the signature type of automatically add or prompt before adding the signature to new items on send.

# SignatureType

Enumerates the signature type of Automatic or Prompt when sending a new item.

## Syntax

```
<SignaturesType>  
<enumeration value="Automatic"/>  
<enumeration value="Prompt"/>
```

# SMimeOperation

Contains a mail message that is signed or encrypted with secure MIME.

## Syntax

```
<SMimeOperation>  
<signed type="boolean"/>  
<encrypted type="boolean"/>
```

## Definitions

### **signed**

Specifies if the mail message is signed.

### **encrypted**

Specifies if the mail message is encrypted.

# Sort

Identifies the type of sort.

## Syntax

```
<Sort>  
<type type="SortType"/>  
<field type="string"/>
```

## Definitions

### type

Specifies whether ascending or descending is the sort order.

### field

Specifies the field.

# SortType

Enumerates the sort type.

## Syntax

```
<SortType>  
<enumeration value="Ascending"/>  
<enumeration value="Descending"/>
```

# SourceType

Specifies the source type.

## Syntax

```
<SourceType>  
<enumeration value="Library"/>  
<enumeration value="User"/>
```

# 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

Is not implemented at this time.

### problems

Specifies any errors.



# StatusTracking

Contains the information to track on a GroupWise sent item.

## Syntax

```
<StatusTracking>  
<extension base="StatusTrackingOptions">  
<attribute name="autoDelete" type="boolean"/>
```

## Definitions

### autoDelete

Specifies to remove items you have sent from your mailbox (after all the recipients have deleted the items and emptied them from the Trash).

# StatusTrackingFlags

Provides a finer definition of the [StatusTracking](#) elements.

## Syntax

```
<StatusTrackingFlags>
  <delivered type="boolean"/>
  <hostDeleted type="boolean"/>
  <deleted type="boolean"/>
  <opened type="boolean"/>
  <full type="boolean"/>
</StatusTrackingFlags>
```

## Definitions

### **delivered**

Notifies if the item is delivered.

### **hostDeleted**

Notifies if the item is deleted from the host.

### **deleted**

Notifies if the item is deleted by the user.

### **opened**

Notifies if the item is opened by the user.

### **full**

Monitors all activity (same as "All" in StatusTracking).

# StatusTrackingOptions

Contains an enumeration for the tracking levels on sent items.

## Syntax

```
<StatusTrackingOptions>  
<enumeration value="None" />  
<enumeration value="Delivered" />  
<enumeration value="DeliveredAndOpened" />  
<enumeration value="All" />
```

## Definitions

### None

Specifies to not track the status for sent items.

### Delivered

Specifies to track only the delivered status on sent items.

### DeliveredAndOpened

Specifies to track the delivered and opened status on sent items.

### All

Specifies to track all statuses on sent items.

# Subscribe

Contains the settings for a notify user. --For GroupWise 8.0 SP1 and later.

## Syntax

```
<Subscribe>  
  <alarm type="boolean"/>  
  <notification type="boolean"/>  
</Subscribe>
```

## Definitions

### alarm

Specifies whether to receive the alarms.

### notification

Specifies whether to receive the notifications.

# SystemFolder

Describes system folders.

## Syntax

```
<SystemFolder>  
<extension base="Folder"/>  
<isSystemFolder type="boolean"/>  
<type type="FolderType"/>  
<acl type="FolderACL"/>  
<isSharedByMe type="boolean"/>
```

## Definitions

### isSystemFolder

Specifies if the folder is a system folder.

### type

Specifies the folder type.

### acl

Specifies the access control list if the folder is shared.

### isSharedByMe

Specifies if the folder is shared to others.

# 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"/>
<timezone type="Timezone">
<taskTimeOffset type="unsingedInt"/>
```

## 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 need to appear before the numeric values. A1, A111, A999, Z1, Z111, and Z999 are acceptable values.

### **completed**

Specifies if the task is completed.

### **timezone**

Specifies the time zone for the task.

### **taskTimeOffset**

Specifies the time offset that the task is due on the due date. The offset is specified from midnight and is entered in seconds.

**--For GroupWise 2014 R2 SP1 and later.**

# TextLines

Contains text lines describing junk mail handling. --For GroupWise 2012 and later.

## Syntax

```
<TextLines>
  <line type="string" maxOccurs="unbounded"/>
</TextLines>
```

## Definitions

### line

Specifies the line of text.

# 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.

# TimezoneList

Lists time zone definitions from the POA.

## Syntax

```
<TimezoneList>  
<timezone type="Timezone" maxOccurs="unbounded"/>
```

# TransferFailedStatus

Indicates to the sender of an item that there was a delivery problem.

## Syntax

```
<TransferFailedStatus>  
<extension base="RecipientStatus">  
<attribute name="FailureReason" type="string"/>
```

## Definitions

### FailureReason

Specifies the reason for the delivery problem.

# 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">
    <name type="string"/>
    <key type="string"/>
    <proxy type="string"/>
    <force type="boolean"/>
  </TrustedApplication>
```

## Definitions

### 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.

### proxy

Specifies the user name of the person to proxy as.

### force

If true, allows a trusted application to log in to a disabled or expired user.

## 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 IDs (up to the at symbol) for both items.

### Syntax

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

# UserContactFolder

Specifies personal address books (PAB) in the folder list.

## Syntax

```
<UserContactFolder>  
<extension base="Folder">  
<folderType type="FolderType"/>  
<addressBook type="uid"/>
```

## Definitions

### **folderType**

Specifies the folder type.

### **addressBook**

Specifies the unique identifier for the address book.

# UserInfo

Contains information about the logged-in user.

## Syntax

```
<UserInfo>
  <element ref="tns:name"/>
  <element ref="tns:email"/>
  <element ref="tns:uuid"/>
  <element name="userid" type="string"/>
  <element name="domain" type="string"/>
  <element name="postOffice" type="string"/>
  <element name="system" type="string"/>
  <element name="fid" type="string"/>
  <element name="recipType" type="tns:RecipientType"/>
  <attribute name="disabled" type="boolean"/>
</UserInfo>
```

## Definitions

### name

Specifies the name of the logged-in user.

### e-mail

Specifies the email address of the logged-in user.

### uuid

Specifies the GroupWise unique identifier for the logged-in user.

### userid

Specifies the userid for the logged-in user.

### domain

Specifies the user's domain.

### postOffice

Specifies the user's post office

### system

Specifies the user's GroupWise system

### fid

Specifies the user's fid.

### recipType

Specifies the recipType for the logged-in user or resource.

**disabled**

Specifies if the user is disabled.

**--For GroupWise 8.0 SP2 and later.**



# UserList

Contains a list of users.

## Syntax

```
<UserList>  
<user type="UserInfo" minOccurs="0" maxOccurs="unbounded"/>
```

# 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"/>
```

# VacationRule

A VacationRule is an abstraction of a basic rule. It represents the vacation rule defined in the GroupWise client. For more information on basic rules, see [Rule](#). This is only available in GroupWise 14.2.1 and later. You must pass 1.08 as the [version](#) or the basic rule definition is returned.

## Syntax

```
<VacationRule>
<extension base="tns:Rule">
<element ref="subject"/>
<message type="tns:MessageBody" />
<replyToExternalUsers" type="xs:boolean"/>
  <startDate type="xs:date"/>
<endDate type="xs:date"/>
</VacationRule>
```

## Definitions

### subject

Specifies the subject line sent in the reply.

### message

Specifies the message body sent in the reply.

### replyToExternalUsers

Specifies whether or not to reply to users external to the GroupWise system.

### startDate

Specifies the start date to send the reply.

### endDate

Specifies the end date to send the reply.

# Version

Represents properties of a particular version of a document in a GroupWise library.

## Syntax

```
<Version>
<extension base="ContainerItem">
<library type="NameAndEmail"/>
<documentNumber type="unsignedInt"/>
<versionCreator type="NameAndEmail"/>
<retrievedBy type="NameAndEmail"/>
<retrievedDate type="dateTime"/>
<versionNumber type="unsignedInt"/>
<versionDescription type="string"/>
<versionStatus type="VersionStatus"/>
<life type="unsignedInt"/>
<ageAction type="AgeAction"/>
<fileSize type="unsignedInt"/>
<filename type="string"/>
```

## Definitions

### **library**

Specifies the document library.

### **documentNumber**

Specifies the document number in the library.

### **versionCreator**

Specifies the user that created the version.

### **retrievedBy**

User that retrieved the document.

### **retrievedDate**

Specifies the date that the version was retrieved.

### **versionNumber**

Specifies the version number of the document.

### **versionDescription**

Specifies the version description of the document.

### **versionStatus**

Specifies the status of the version.

### **life**

Specifies the document life in days. After the life in days, the ageAction occurs.

**ageAction**

Specifies the action that occurs to a document when the life of the document reaches the life value in days: archive, delete, etc.

**fileSize**

Specifies the size of the document.

**filename**

Specifies the extension of the file.

# VersionEvent

Contains the activity logs for a particular version of a document. For example, when the access list on a item is modified, a VersionEvent is created.

## Syntax

```
<VersionEvent>
<extension base="ContainerItem">
<library type="NameAndEmail" />
<documentNumber type="unsignedInt" />
<versionNumber type="unsignedInt" />
<creator type="NameAndEmail" />
<event type="VersionEventType" />
<eventNumber type="unsignedInt" />
<filename type="string" />
```

## Definitions

### library

Specifies the document library.

### documentNumber

Specifies the document number in the library.

### versionNumber

Specifies the document version number.

### creator

Specifies the creator of the document version

### event

Specifies the document version event type. For example, the document was checkOut.

### eventNumber

Specifies the number of the event.

### filename

Specifies the extension of the file.

# VersionEventType

Contains an enumeration of version event types (document actions or events such as archive or checkout).

## Syntax

```
<VersionEventType>
<enumeration value="archive" />
<enumeration value="checkIn" />
<enumeration value="checkOut" />
<enumeration value="copyIn" />
<enumeration value="copyOut" />
<enumeration value="copyVersion" />
<enumeration value="created" />
<enumeration value="deleteVersion" />
<enumeration value="endAccess" />
<enumeration value="resetStatus" />
<enumeration value="restore" />
<enumeration value="retrieve" />
<enumeration value="securityModified" />
<enumeration value="versionDownloaded" />
<enumeration value="viewed" />
<enumeration value="unknown" />
```

# VersionStatus

Contains an enumeration of the version status information.

## Syntax

```
<VersionStatus>
<enumeration value="available"/>
<enumeration value="checkedOut"/>
<enumeration value="inUse"/>
<enumeration value="deleted"/>
<enumeration value="archived"/>
<enumeration value="massInUse"/>
<enumeration value="unavailable"/>
```

## Definitions

### **available**

Specifies that the document version is available.

### **checkedOut**

Specifies that the document version has been checked out.

### **inUse**

Specifies that the document version is in use.

### **deleted**

Specifies that the document version has been deleted.

### **archived**

Specifies that the document version has been archived.

### **massInUse**

Specifies that the document is in use by several users.

### **unavailable**

Specifies that the document version is unavailable.



# View

Contains a space-separated list of fields to retrieve in a getItem request. For more information see [Filters and Views](#).

## Syntax

```
<View>  
<list itemType="string"/>
```

# ViewType

Enumerates the view type in the display settings.

## Syntax

```
<ViewType>  
<enumeration value="details"/>  
<enumeration value="largeIcons"/>  
<enumeration value="smallIcon"/>  
<enumeration value="list"/>  
<enumeration value="discussion"/>  
<enumeration value="calendar"/>  
<enumeration value="tasklist"/>  
<enumeration value="panel"/>  
<enumeration value="addressCards"/>
```

# Visibility

Specifies where an address book item is visible.

## Syntax

```
<Visibility>
  <enumeration value="PostOffice"/>
  <enumeration value="Domain"/>
  <enumeration value="System"/>
  <enumeration value="Unlisted"/>
  <enumeration value="Unknown"/>
</Visibility>
```

# WebAccessSettings

Contains settings needed for WebAccess. --For GroupWise 2012 and later.

## Syntax

```
<WebAccessSettings>
  <extension base="Item">
    <timezone type="base64Binary"/>
    <timezoneId type="int"/>
    <timezoneMatches type="base64Binary"/>
    <language type="language"/>
    <general type="base64Binary"/>
  </WebAccessSettings>
```

# 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"/>
```

