

WebAccess Customization

GroupWise® Software Developer Kit

November 2012

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to <http://www.novell.com/info/exports/> (<http://www.novell.com/info/exports/>) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 1993-2007, 2012 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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

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

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide	9
1 Overview	11
1.1 Accessing GroupWise from a Web Browser	12
1.1.1 Servlet Communication	12
1.1.2 WebAccess Providers	12
1.2 Changing the Company Logo	12
1.3 Modifying the WebAccess User Interface	13
1.4 Adding Other Services or Databases	13
2 Template Concepts	15
2.1 Template Overview	15
2.2 Scripting Keywords and Constructs	17
2.3 Functions	18
2.4 Comments	18
2.5 Replaceable Parameters	19
2.6 Dates	20
2.7 Arrays	20
2.8 Localization	20
3 Template Tasks	23
3.1 Review Prerequisites	25
3.2 Changing the Company Logo and Login Files	25
3.2.1 Modify Files to Change Company Graphics	25
3.2.2 Include Logo Graphic Files in the Image Directory	26
3.2.3 View on Web Server	26
3.3 Modifying the Main Message Windows	27
3.3.1 Mail Message Window	28
3.4 Modifying the Compose Windows	29
3.5 Modifying the Address Book Windows	30
3.6 Modifying the Calendar Windows	31
3.6.1 Daily Calendar	31
3.6.2 Weekly Calendar	33
3.6.3 Monthly Calendar	35
3.7 Modifying the Document Management Windows	37
4 Template Keywords	39
4.1 ABORT	39
4.2 ALIAS	39
4.3 COOKIE	41
4.4 DEC	41
4.5 DUMPALL	41
4.6 HEADER	42
4.7 IF...ELSEIF...ELSE...ENDIF	
IFDEF...ELSEIFDEF	42

4.8	IGNOREWHITE and NOIGNOREWHITE	43
4.9	INC	43
4.10	INCLUDE	43
4.11	NOOUTPUT	44
4.12	OUTPUT	44
4.13	SET	44
4.14	SETGLOBAL	44
4.15	STRING	45
4.16	STRINGTABLE	45
4.17	VAR	45
4.18	WHILE...BREAK...CONTINUE...ENDWHILE	46

5 Template Functions 47

addToDate	49
createDate	50
date	51
daysInMonth	53
endsWith	54
find	55
findLast	56
firstWeekDayOfMonth	57
firstDayOfWeek	58
formatMessage	59
getHeader	60
length	61
linkEncodedURLs	62
linkURLs	63
listToArray	65
listToSortedArray	66
noScript	67
replaceWith	68
startsWith	69
substring	70
today	71
todayTZ	72
toDisplay	73
toHTML	74
toHTMLExt	75
toLower	76
toQuote	77
toScript	78
toSpellerText	79
toUpper	80
toWMLExt	81
toXML	82
trim	83
urlEncode	84
xmlDecode	85
xmlEncode	86

6	Servlet Concepts	87
6.1	Configuration Settings	87
6.2	Java Packages	92
6.3	Providers	93
6.3.1	WebAccess GroupWise Provider	93
6.3.2	WebAccess LDAP Provider	93
6.4	Localization and Internationalization	94
7	Servlet Tasks	95
7.1	Configuring the Servlet	95
7.2	Calling the Servlet	95
7.3	Specifying a Provider	96
7.4	Specifying the Action	96
7.5	Specifying a Template	97
7.5.1	Specifying the Template If an Error Occurs	97
7.5.2	Adjusting the Cache Size	97
8	Servlet Methods	99
	User.Login	100
	User.Logout	101
9	Provider Tasks	103
9.1	Calling the GroupWise Provider	103
9.2	Communicating With the Servlet	103
9.3	Specifying a Template	103
9.3.1	Specifying a Template If an Error Occurs	104
10	Provider Methods	105
10.1	Attachment	106
	Attachment.Save	107
	Attachment.View	108
10.2	Calendar	109
	Calendar.Search	110
10.3	Compose	114
	Compose.Action	115
	Compose.Attach	123
	Compose.Attach.Delete	126
	Compose.BC	129
	Compose.BusySearch	130
	Compose.Cancel	133
	Compose.CC	134
	Compose.ChangeTo.xxx	135
	Compose.Delegate	136
	Compose.Directory	139
	Compose.FolderSel	140
	Compose.Forward	141
	Compose.Reply	144
	Compose.SaveDraft	147
	Compose.Send	148
	Compose.SendOptions.Modify	149
	Compose.SendOptions.Read	151
	Compose.To	153

10.4	Directory	156
	Directory.Accumulate.Group	157
	Directory.Accumulator.Add.BC	158
	Directory.Accumulator.Add.CC	159
	Directory.Accumulator.Add.To	160
	Directory.Accumulator.Delete	161
	Directory.Accumulator.List	162
	Directory.Action	163
	Directory.CollapseGroup	165
	Directory.CopyEntries	166
	Directory.Create	167
	Directory.CreateEntry	168
	Directory.CreateGroup	171
	Directory.Delete	174
	Directory.DeleteEntry	175
	Directory.GetLastSearch	176
	Directory.HoldEntries	178
	Directory.Info	179
	Directory.InfoEntry	180
	Directory.List	182
	Directory.Modify	183
	Directory.ModifyEntry	184
	Directory.ModifyGroup	187
	Directory.MoveEntries	189
	Directory.SaveAs	190
	Directory.Search	191
	Directory.ViewEntry	195
10.5	Document	196
	Document.Profile	197
	Document.Save	198
	Document.View	199
	Document.ViewNative	200
10.6	Find	201
	Find.Action	202
	Find.Beginning	203
	Find.First	204
	Find.NewUpdate	208
	Find.Next	209
	Find.Prev	210
	Find.StopSearch	211
	Find.Update	212
10.7	Folder	213
	Directory.Cancel	214
	Directory.GetRecipients	215
	Folder.Action	217
	Folder.Create	218
	Folder.Delete	219
	Folder.Directory	220
	Folder.Empty	221
	Folder.Expand	222
	Folder.List	223
	Folder.ListChild	225
	Folder.ListShare	227
	Folder.Open	228
	Folder.Share	231
10.8	Library	233
	Library.List	234
	Library.Search	235
10.9	Item	237

Item.Accept	238
Item.AcceptAll	239
Item.AcceptSharedAddressBook	240
Item.Action	241
Item.CheckList.Bottom	243
Item.CheckList.Down	244
Item.CheckList.Save	245
Item.CheckList.Top	246
Item.CheckList.Up	247
Item.Complete	248
Item.Compose	249
Item.Decline	250
Item.DeclineAll	251
Item.Delete	252
Item.MarkRead	253
Item.Move	254
Item.Next	255
Item.Previous	256
Item.Properties	257
Item.Properties.Read	259
Item.Properties.ReadInfo	262
Item.Properties.ReadNext	265
Item.Purge	266
Item.Read	267
Item.ReadChild	271
Item.ReadLater	272
Item.ReadParent	273
Item.Retract	274
Item.RetractAll	275
Item.Undelete	276
10.10 Miscellaneous	277
XKI.getDRNList	278
10.11 Proxy	279
Directory.Cancel	280
Directory.GetRecipients	281
Proxy.Action	282
Proxy.AddAccess	283
Proxy.CloseListAccess	285
Proxy.CloseListHistory	286
Proxy.DeleteAccess	287
Proxy.DeleteHistory	288
Proxy.Directory	289
Proxy.ListAccess	290
Proxy.ListHistory	291
Proxy.Login	292
Proxy.ModifyAccess	293
Proxy.ReadAccess	295
10.12 Rules	296
Directory.GetRecipients	297
Rule.Action	298
Rule.Cancel	299
Rule.Create	300
Rule.CreateType	302
Rule.Delete	303
Rule.Directory	304
Rule.Enable	306
Rule.List	307
Rule.Modify	308
Rule.Read	310
10.13 Security	311

Security.ChangePassword	312
10.14 Send Options	313
SendOptions.Modify	314
SendOptions.Read	317
10.15 Settings	319
Settings.General.Modify	320
Settings.Get	321
Settings.Set	322
10.16 Signature	323
Signature.Modify	324
Signature.Read	325
10.17 Time Zone	326
TimeZone.List	327
TimeZone.Modify	328
TimeZone.Read	329
10.18 User Authentication	330
User.Login	331
User.Logout	333
11 LDAP Provider Tasks	335
11.1 Calling the LDAP Provider	335
11.2 Calling the Servlet	335
11.3 Adding Additional LDAP Directories	335
11.4 Specifying a Template	336
12 LDAP Provider Methods	337
Directory.Action	338
Directory.List	339
Directory.Search	340
13 WebPublisher Provider Tasks	343
13.1 Calling the WebPublisher Provider	343
13.2 Communicating With the Servlet	344
13.3 Specifying a Template	344
14 WebPublisher Provider Methods	345
Document.Profile	346
Document.Save	347
Document.View	348
Library.List	349
Library.Search	350
15 Revision History	353

About This Guide

GroupWise WebAccess allows you to access your GroupWise mailbox and calendar from the Internet. GroupWise WebAccess Customization lets you modify GroupWise WebAccess.

IMPORTANT: The features in GroupWise WebAccess customization work with GroupWise 8. They do not work with GroupWise 2012 and later versions because WebAccess was redesigned in GroupWise 2012.

This guide shows you how to customize GroupWise WebAccess to meet the needs of your GroupWise users and contains the following sections:

- ♦ Chapter 1, "Overview," on page 11
- ♦ Chapter 2, "Template Concepts," on page 15
- ♦ Chapter 3, "Template Tasks," on page 23
- ♦ Chapter 4, "Template Keywords," on page 39
- ♦ Chapter 5, "Template Functions," on page 47
- ♦ Chapter 6, "Servlet Concepts," on page 87
- ♦ Chapter 7, "Servlet Tasks," on page 95
- ♦ Chapter 8, "Servlet Methods," on page 99
- ♦ Chapter 9, "Provider Tasks," on page 103
- ♦ Chapter 10, "Provider Methods," on page 105
- ♦ Chapter 11, "LDAP Provider Tasks," on page 335
- ♦ Chapter 12, "LDAP Provider Methods," on page 337
- ♦ Chapter 13, "WebPublisher Provider Tasks," on page 343
- ♦ Chapter 14, "WebPublisher Provider Methods," on page 345
- ♦ Chapter 15, "Revision History," on page 353

Audience

This guide is intended for GroupWise developers.

Feedback

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

Additional Documentation

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

1 Overview

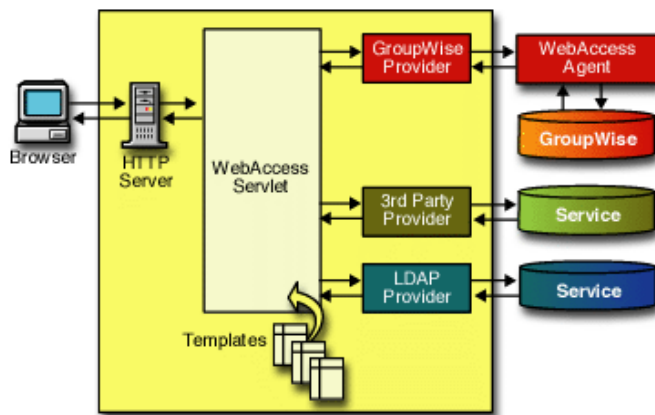
GroupWise WebAccess allows you to access your GroupWise mailbox and calendar from the Internet.

This section covers the following topics:

- ♦ [Section 1.1, “Accessing GroupWise from a Web Browser,”](#) on page 12
- ♦ [Section 1.2, “Changing the Company Logo,”](#) on page 12
- ♦ [Section 1.3, “Modifying the WebAccess User Interface,”](#) on page 13
- ♦ [Section 1.4, “Adding Other Services or Databases,”](#) on page 13

The underlying architecture of GroupWise 8 WebAccess consists primarily of two servers (a two-box solution), one for the GroupWise WebAccess Agent and one for HTTP and the servlet, which are connected through TCP/IP. Data security can be provided by a firewall located between the two servers, as shown in the following figure:

Figure 1-1 Architecture of GroupWise WebAccess



The GroupWise WebAccess Agent is available for NetWare and Windows. The servlet server is available for NetWare, Windows, and UNIX.

By installing the GroupWise WebAccess Agent program (GWINTER) on a single GroupWise domain, you can access any post office in your system from any HTTP server that is running. If you are running an Intranet or if security is not a concern, you can run GWINTER on the same machine as the HTTP server and the servlet (a one box solution).

For this document, GWINTER will indicate both GWINTER.NLM and GWINTER.EXE.

1.1 Accessing GroupWise from a Web Browser

Novell WebAccess GroupWise Provider is a component of GroupWise WebAccess that interfaces with GroupWise to retrieve information and perform operations. This section describes the commands and parameters that can be used in URLs and HTML forms to allow access to GroupWise from a browser.

This section covers the following topics:

- ♦ [Section 1.1.1, “Servlet Communication,” on page 12](#)
- ♦ [Section 1.1.2, “WebAccess Providers,” on page 12](#)

1.1.1 Servlet Communication

Actions and their associated parameters are processed by the WebAccess servlet. The servlet gets installed when you install GroupWise WebAccess. To call the servlet, create a URL that references the IP address or host name of the computer where the servlet was installed (which is the machine running your HTTP server) and specify /servlet/webacc. For example

```
http://myserver.mycompany.com/servlet/webacc
```

1.1.2 WebAccess Providers

To call the provider, include in the URL or a hidden form field, the word “action” accompanied by the name of the action to perform, such as User.Login. Also include in the URL or other form fields, the name of the necessary parameters and their values.

The name of the GroupWise provider is GWAP. When WebAccess is installed, GWAP is specified as the default provider. Since it is the default provider, it is not necessary to specify the name of the provider in the URL or forms. If the default provider is changed (in webacc.cfg), it becomes necessary to include the Provider.name parameter with all requests so that the servlet calls the GroupWise provider.

1.2 Changing the Company Logo

GroupWise WebAccess is modifiable for your environment. You can change the company logo and graphics to fit your company needs. You can also make other changes to the user interface, see [“Template Concepts” on page 15](#).

To change the WebAccess user interface, see the following information:

- ♦ [Section 3.1, “Review Prerequisites,” on page 25](#)
- ♦ [Section 3.2.1, “Modify Files to Change Company Graphics,” on page 25](#)
- ♦ [Section 3.2.2, “Include Logo Graphic Files in the Image Directory,” on page 26](#)
- ♦ [Section 3.2.3, “View on Web Server,” on page 26](#)

1.3 Modifying the WebAccess User Interface

The WebAccess user interface is made up of files referred to as template files. These template files contain HyperText Templates (.htt), include files (.inc), and applet files (.applet). You can modify these template files to change the WebAccess user interface to meet your needs.

To modify the user interface to meet your needs, see the following information:

- ♦ [Section 3.1, “Review Prerequisites,” on page 25](#)
- ♦ [Chapter 2, “Template Concepts,” on page 15](#)
- ♦ [Chapter 3, “Template Tasks,” on page 23](#)
- ♦ [Chapter 4, “Template Keywords,” on page 39](#)
- ♦ [Chapter 5, “Template Functions,” on page 47](#)

1.4 Adding Other Services or Databases

Novell WebAccess Servlet is a flexible framework to generate a Web interface for an application. The Servlet architecture consists of the servlet, providers, and templates. For more information, see the architecture graphic in the [“Overview” on page 11](#).

The servlet receives requests from a browser or other HTTP client, and handles those requests by calling the appropriate provider. The provider does the requested work and returns the results. The results are combined with a template to produce a text-based document (HTML, XML, etc.), which is then sent back to the browser or HTTP client.

To add other services or databases, review the following information:

- ♦ WebAccess Servlet
 - ♦ [Chapter 6, “Servlet Concepts,” on page 87](#)
 - ♦ [Chapter 7, “Servlet Tasks,” on page 95](#)
 - ♦ [Chapter 8, “Servlet Methods,” on page 99](#)
- ♦ GroupWise Providers
 - ♦ [Chapter 9, “Provider Tasks,” on page 103](#)
 - ♦ [Chapter 10, “Provider Methods,” on page 105](#)
- ♦ LDAP Providers
 - ♦ [Chapter 11, “LDAP Provider Tasks,” on page 335](#)
 - ♦ [Chapter 12, “LDAP Provider Methods,” on page 337](#)
- ♦ WebPublisher Servlet/Provider
 - ♦ [Chapter 13, “WebPublisher Provider Tasks,” on page 343](#)
 - ♦ [Chapter 14, “WebPublisher Provider Methods,” on page 345](#)
- ♦ Creating Your Own Providers
 - ♦ [Javadoc Reference for the Provider Interface \(see \[Section 6.3, “Providers,” on page 93\]\(#\)\)](#)

2 Template Concepts

Templates give you the ability to generate HTML or other text-based documents that incorporate data provided by the servlet. By modifying the templates you can customize WebAccess to meet your organization's needs. Templates consist of text (such as HTML) embedded with special template commands. While this document uses HTML examples, note that XML or any other text-based documents can be generated.

This section covers the following topics:

- ♦ [Section 2.1, "Template Overview," on page 15](#)
- ♦ [Section 2.2, "Scripting Keywords and Constructs," on page 17](#)
- ♦ [Section 2.3, "Functions," on page 18](#)
- ♦ [Section 2.4, "Comments," on page 18](#)
- ♦ [Section 2.5, "Replaceable Parameters," on page 19](#)
- ♦ [Section 2.6, "Dates," on page 20](#)
- ♦ [Section 2.7, "Arrays," on page 20](#)
- ♦ [Section 2.8, "Localization," on page 20](#)

2.1 Template Overview

Templates are text-based documents such as HTML or XML embedded with keywords and scripting constructs. When the templates are processed, the keywords and scripting constructs are replaced to produce the final document.

The following template generates an HTML page that gives information about the browser that is requesting the page:

```
<HTML>
<HEAD><TITLE>Browser Information Sample</TITLE></HEAD>
<BODY>
<H1>Browser Information</H1>
Browser Name: {VAR Browser.name}<BR>
Major Version: {VAR Browser.mjrVer}<BR>
Minor Version: {VAR Browser.mnrVer}<BR>
```

This will generate an HTML page based on the browser that requests the page. The resulting HTML is as follows:

```
<HTML>
<HEAD><TITLE>Browser Information Sample</TITLE></HEAD>
<BODY>
<H1>Browser Information</H1>
Browser Name: NSN<BR>
Major Version: 4<BR>
Minor Version: 05<BR>
```

A Browser might present this page as follows:

```
Browser Information
Browser Name: NSN
Major Version: 4
Minor Version: 05
```

The following table lists the templates most likely needed for modification of the user interface. Use a editor that is UTF-8 compatible to edit any of these files. For more information and graphical representations of each file, see [“Template Concepts” on page 15](#).

Template File Name	Description	Includes Other Template Files
login.htm	Login frameset and includes logo graphic files.	gbltop.inc popout.inc error.inc
webacc.htm	Top level frameset in which all other windows appear.	gbltop.inc
gbltop.inc	Common include file for settings used by all templates.	None
caption.htm	Content for the caption window at the top of the user interface (top frame in webacc.htm).	gbltop.inc popout.inc
maindoc.htm	Frameset containing all windows except the caption window (bottom frame in webacc.htm).	gbltop.inc
features.htm	Content for the left most window of the user interface containing the navigation buttons or links (i.e. compose an email, access address book, access calendar, access document management). (Left frame in maindoc.htm.)	gbltop.inc
workspace.htm	Frameset containing the folder list and messages list windows. (Right frame in maindoc.htm)	gbltop.inc
addrbook.htm	Frameset containing the HTML version of the Address book, or page containing the Java version of the address book.	gbltop.inc addrbook.applet
calset.htm	Frameset containing calendar navigation and calendar views.	None
calendar.htm	Day Calendar view. Appears in same window as the messages list.	gbltop.inc popout.inc error.inc fldrmove.inc
folders.htm	Content for the folder list.	gbltop.inc boldlink.inc error.inc fldricon.inc
msglist.htm	Content containing the messages in a folder.	gbltop.inc popout.inc error.inc fldrmove.inc

Template File Name	Description	Includes Other Template Files
send.htm	Content to compose/foward/reply/delegate a message.	gbldtop.inc popout.inc captbar.inc addattch.inc speller.applet sendres.inc sendoptions.inc send.inc
addrsrch.htm	Frame containing address book search query in the HTML version of the address book.	gbldtop.inc popout.inc error.inc
addreslt.htm	Frame containing the result list of an address book search for the HTML version of the address book.	gbldtop.inc popout.inc error.inc
liblist.htm	Frame containing the library list and specify a document query.	gbldtop.inc error.inc
doclist.htm	Frame containing results list of a document search.	gbldtop.inc popout.inc error.inc
docprop.htm	Frame containing the properties of document.	gbldtop.inc captbar.inc docprop.inc
addrbook.applet	Include file that runs the address book applet.	None
speller.applet	Include file that runs the spell check applet	None
calendar.applet	Include file that runs the Java applet version of the calendar.	None

2.2 Scripting Keywords and Constructs

Templates use braces ({}) to identify keywords and constructs embedded in the document. For example, {VAR User.name} identifies the keyword VAR followed by the name of a variable which the VAR keyword requires.

If an open brace is not followed by a keyword, it is not considered a part of a keyword and is treated as any other character in the document. White space such as spaces, tabs, carriage returns and line feeds can occur anywhere between the braces, keywords and parameters. The only exception is the comment keyword (!) which requires the comment character to immediately follow the opening brace: {! This is a comment. }.

Keywords are case sensitive. Keywords can be nested with other keywords. However, if the nested keyword is inside a quoted string, the nested keyword will not be evaluated but will be considered a part of the string.

The following sample:

```
{SET test = "TEST"}
{SET a = "This is a "{VAR test}}
{VAR a}
```

Produces this output:

```
This is a TEST
```

The following sample:

```
{SET test = "TEST"}
{SET a = "This is a {VAR test}"}
{VAR a}
```

Produces this output:

```
This is a {VAR TEST}
```

2.3 Functions

Functions cause special processing to occur. They usually appear within a SET, STRING, or VAR keyword.

```
{SET TisATest="this is a test"}
{VAR substring(TisATest, 8, 6)} {! output "a test"}
```

Function names are case sensitive.

Functions can be used in expressions as long as the expression is parenthesized:

```
{SET tomorrow = (today() + 86400000)}
```

Arithmetic expressions as function parameters must be enclosed in parentheses:

```
{SET curDay = today() }
```

Tomorrow is {VAR date((curDay + 86400000), {STRING fullDate})}

Functions can serve as parameters to other functions:

```
{VAR date( today(), {STRING fullDate})}
```

Variables and names of strings in STRING TABLES cannot have the name of a function.

```
{SET date = today() } {! Illegal. date is the name of a function. }
```

2.4 Comments

You can insert a comment in a template by enclosing the text of the comment in braces, with an exclamation mark immediately following the opening brace. Other braces can be nested within the comment such that when an opening brace is encountered inside a comment, the next closing brace encountered will not close the comment. This allows you to easily comment out sections of the template that may contain other keywords or comments.

Comments are removed from the final document being produced.

```
<HTML>
<BODY>
{! This section has been commented out
  {SET a=1}
  <H1>Page Number {VAR a}</H1>
}
  <H1>Welcome to My Page</H1>
</BODY>
</HTML>
```

The preceding example will produce the following HTML text document:

```
<HTML>
<BODY><H1>Welcome to My Page</H1></BODY>
</HTML>
```

2.5 Replaceable Parameters

A special syntax is available for specifying values containing replaceable parameters. A replaceable parameter appears in a value as a number between braces, such as {0}. A replaceable parameter can be embedded within a string in a string table, or in other variables referenced with the VAR keyword. The replaceable parameter will be replaced with the appropriate value when the document is generated based on the parameters used in the template. For example, in a string table there could be an entry such as:

```
srchResults=You have {0} messages.
```

A template would refer to this string as follows:

```
{STRING srchResults {VAR Item.count}}
```

So that srchResults is the name of the string that should be added to the final document. {Item.count} is the parameter that replaces {0} in the string. Therefore, if Item.count had a value of 10, the resulting string would be:

```
You have 10 messages.
```

If there are replaceable parameters, the number found in the parameter refers to the position of the parameters in the template. {0} refers to the first replaceable parameter, {1} refers to the second parameter, etc.

```
# String Table entry:
# {0} is the first parameter in the template,
# {1} is the second parameter in the template,
# {2} is the third parameter in the template.
ThreeValues=Value 2 is {1}, value 1 is {0}, and value 3 is {2}.
{! Template usage }
{SET v1="One" }
{SET v2="Two" }
{SET v3="Three" }
{STRING threeValues {VAR v1} {VAR v2} {VAR v3}}
```

This example outputs the following:

```
Value 2 is Two, value 1 is One, and value 3 is Three.
```

2.6 Dates

Date formats can be specified in string tables or as variables in a template. In the templates, dates are represented as numbers that are translated into various date formats using the date function (see the section on functions).

The following sample date formats could appear in a template or STRINGTABLE.

```
# 10/27/98
numericDate=M/d/yy
# 2:30 PM
meetingTime=h:mm a
# Friday, November 13, 1998
fullDate=EEEE, MMMM d, YYYY
#September 1998
monthDate=MMMM YYYY
```

Assuming these formats exist in a string table, a template would use one of these formats as a parameter to the date function as follows:

```
{VAR date(apptDate, {STRING fullDate})}
```

This results in the following string being inserted:

```
Friday, July 16, 1999
```

2.7 Arrays

An array is a list of variables that can be indexed by a numeric variable. For example, an item could have its attachments in an array. All the attachments names could be listed by indexing the array:

```
{SET i = 1}
{WHILE i <= Item.Attach.count}
{VAR Item.Attach[i].name}
{INC i}
{ENDWHILE}
```

Array indexes can be specified more than once.

```
Item[i].subItem[j].value
Table[i][j]
```

When creating arrays in String Tables, use dots instead of brackets:

```
# refer to these in the templates with brackets (Month[1])Month.1 = JanuaryMonth.2
= Februaryetc.
```

2.8 Localization

Localization is facilitated by the use of string tables (see STRINGTABLE).

One method of accomplishing localization is to create a base string table that contains non-translatable strings. Give this string table a file name consisting of a base file name followed by .properties. Then for each language, create a string table file which uses the base file name, followed by _xx.properties where xx is replaced by the two letter language code for the language the string table is used for.

For example:

- ◆ `FramesResources.properties` contains non-translatable strings such as background colors. It also contains default strings to be used if a language is specified for which no string table exists.
- ◆ `FramesResources_en.properties` contains the strings that are used for English.
- ◆ `FramesResources_fr.properties` contains the strings that are used for French.

In the build process, you may want to concatenate the contents of `FramesResources_en.properties` to `FramesResources.properties`. This makes English the default language, which is useful in the case where a non-existent language code is specified.

3 Template Tasks

Templates give you the ability to generate HTML or other text-based documents that incorporate data provided by the servlet. By modifying the templates you can customize WebAccess to meet your organization's needs. Templates consist of text (such as HTML) embedded with special template commands. While this document uses HTML examples, note that XML or any other text-based documents can be generated.

The GroupWise WebAccess interface is made up of HyperText Template (.htt), include files (.inc), and applet files (.applet). These files are HTML files that use [“Template Keywords” on page 39](#).

This section covers the following:

- ♦ [Section 3.1, “Review Prerequisites,” on page 25](#)
- ♦ [Section 3.2, “Changing the Company Logo and Login Files,” on page 25](#)
- ♦ [Section 3.3, “Modifying the Main Message Windows,” on page 27](#)
- ♦ [Section 3.4, “Modifying the Compose Windows,” on page 29](#)
- ♦ [Section 3.5, “Modifying the Address Book Windows,” on page 30](#)
- ♦ [Section 3.6, “Modifying the Calendar Windows,” on page 31](#)
- ♦ [Section 3.7, “Modifying the Document Management Windows,” on page 37](#)

The interface has some *Java-enhanced* features that are applied with the use of the applets. These features include the address book, calendar, and spell checker. The features are available in both HTML and Java-enhanced modes. The default setup is to use the Java-enhanced mode. The user is also able to toggle back and forth between HTML and Java-enhanced modes from the interface of the these features. Whichever was the last mode (HTML or Java) is remembered the next time the user logs in.

GroupWise WebAccess utilizes HTML frames. The webaccess.htt file (code snippet below) is a base HTML frame file that calls other HTML files with template extensions .htt or .inc, such as caption.htt.

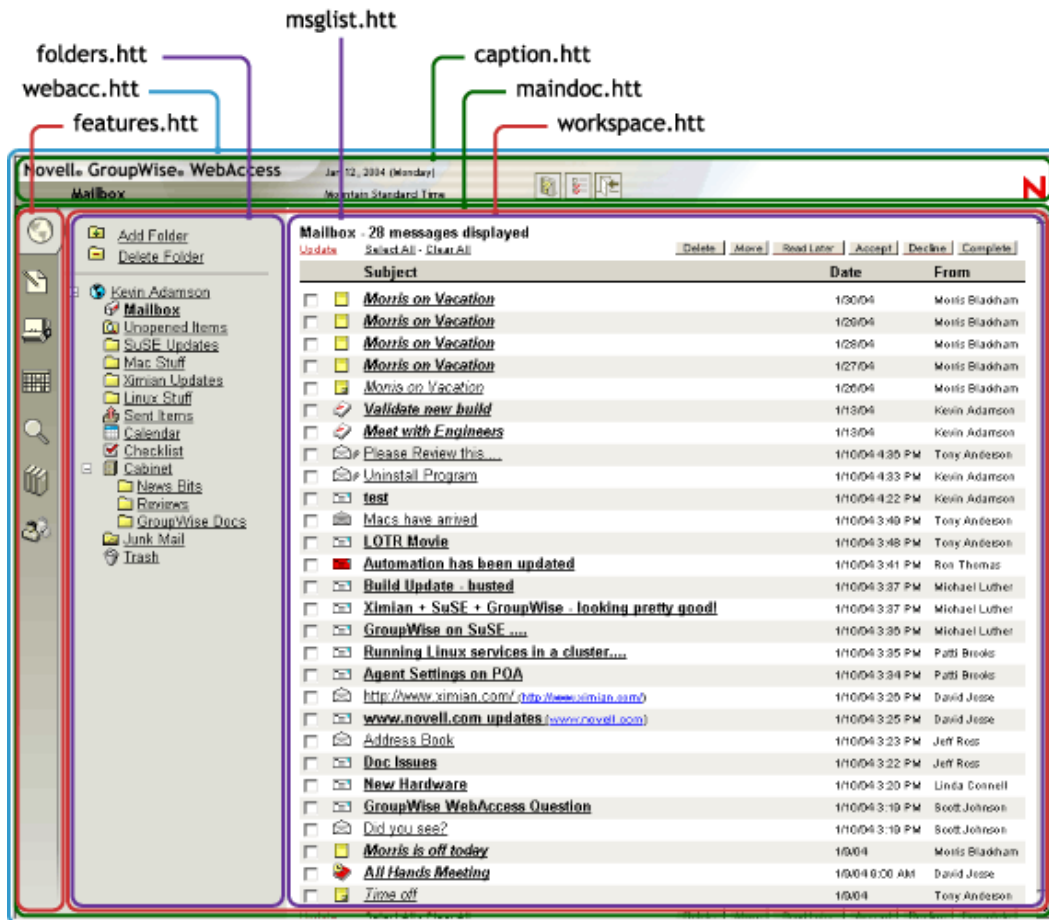
```

<FRAMESET rows="{VAR captsize},*" border="0">
  <FRAMESET cols="*,0" border="0">
    <FRAME src="/servlet/webacc?User.context={VAR
      User.context}&merge=caption"
      FrameBorder=0
      MarginHeight=0
      MarginWidth=0
      Scrolling="No"
      Name="caption">
    <FRAME src="/servlet/webacc?User.context={VAR
      User.context}&merge=mcblank"
      FrameBorder=0
      MarginHeight=0
      MarginWidth=0
      Scrolling="No"
      Name="javamc">
  </FRAMESET>
<FRAME src="/servlet/webacc?User.context={VAR
  User.context}&merge=maindoc"
  FrameBorder=0
  MarginHeight=0
  MarginWidth=0
  Scrolling="No"
  Name="maindoc">
</FRAMESET>

```

The following shows the interface of the previous code as well as nested information pulled from other files.

Figure 3-1 Frames With Embedded Files



To change the interface, you need access rights on the Web server to the template files as outlined in [“Review Prerequisites” on page 25](#).

The following sections show graphics of the interface in relation to the template files and how the files are called and nested from other files. This will help you find the file you need to modify to change the interface to meet your needs.

For a table that contains a brief description of template files and references to related .htt, .inc, or .applet files, see [“Template Concepts” on page 15](#).

3.1 Review Prerequisites

You need the following to change the company logo or modify the user interface:

- ♦ Access and write rights to modify the template files on the Web server.
- ♦ Your company logo graphic files in an Internet accessible format (i.e. .GIF or .JPEG)
- ♦ Location path to template files. On Netscape Enterprise Server for NetWare, the default install location path is

```
z:\java\servlets\com\novell\webaccess\templates\frames
```

- ♦ Location path to graphic files. On Netscape Enterprise Server for NetWare, the default install location path is

```
z:\novonyx\suitespot\docs\com\novell\webaccess\images
```

TIP: If GroupWise WebAccess was installed to a different path or you are using a Web server other than the Netscape Enterprise Server for NetWare, check with your network administrator for the template or graphic file location or search the Web server for the login.htt file. If this file is found in the GRPWISE/SOFTWARE directory, it is the file used for network installations. It is not the file to modify to change on your Web server.

3.2 Changing the Company Logo and Login Files

To change the logo or make the interface better meet your needs, the following sections outline specific steps, including:

- ♦ [Section 3.2.1, “Modify Files to Change Company Graphics,” on page 25](#)
- ♦ [Section 3.2.2, “Include Logo Graphic Files in the Image Directory,” on page 26](#)
- ♦ [Section 3.2.3, “View on Web Server,” on page 26](#)

3.2.1 Modify Files to Change Company Graphics

To change the company logo and other associated attributes throughout the WebAccess interface, you need to modify a file named Customization.properties, which is located in the templates directory. The templates directory is specified by the Templates.path key in the webacc.cfg configuration file.

There are several attributes of the image that can be customized, such as the mouse over text and the URL link that is used when a user clicks on the image. To see a complete list of the available attributes, see the Customization.properties file.

The Customization.properties file also includes attributes to easily override and customize the default color scheme that is used by WebAccess.

3.2.2 Include Logo Graphic Files in the Image Directory

Put company logo graphic files in image directory (See [“Review Prerequisites” on page 25](#)).

If you are not using the install location image directory, replace the following keyword syntax in the template files with the relative path to your Web server's graphic directory:

```
{VAR Templates.Images.URL}
```

3.2.3 View on Web Server

You might need to clear the cache on the server to see your changes.

To avoid having to restart or unload Java every time you make changes, go into the webacc.cfg file and set Templates.Cache.enable to false. The changes to webacc.cfg will not take affect until you restart the servlet. Once restarted, you can edit the templates and include files, then login to GroupWise WebAccess to see your changes without restarting the servlet after every edit.

3.3 Modifying the Main Message Windows

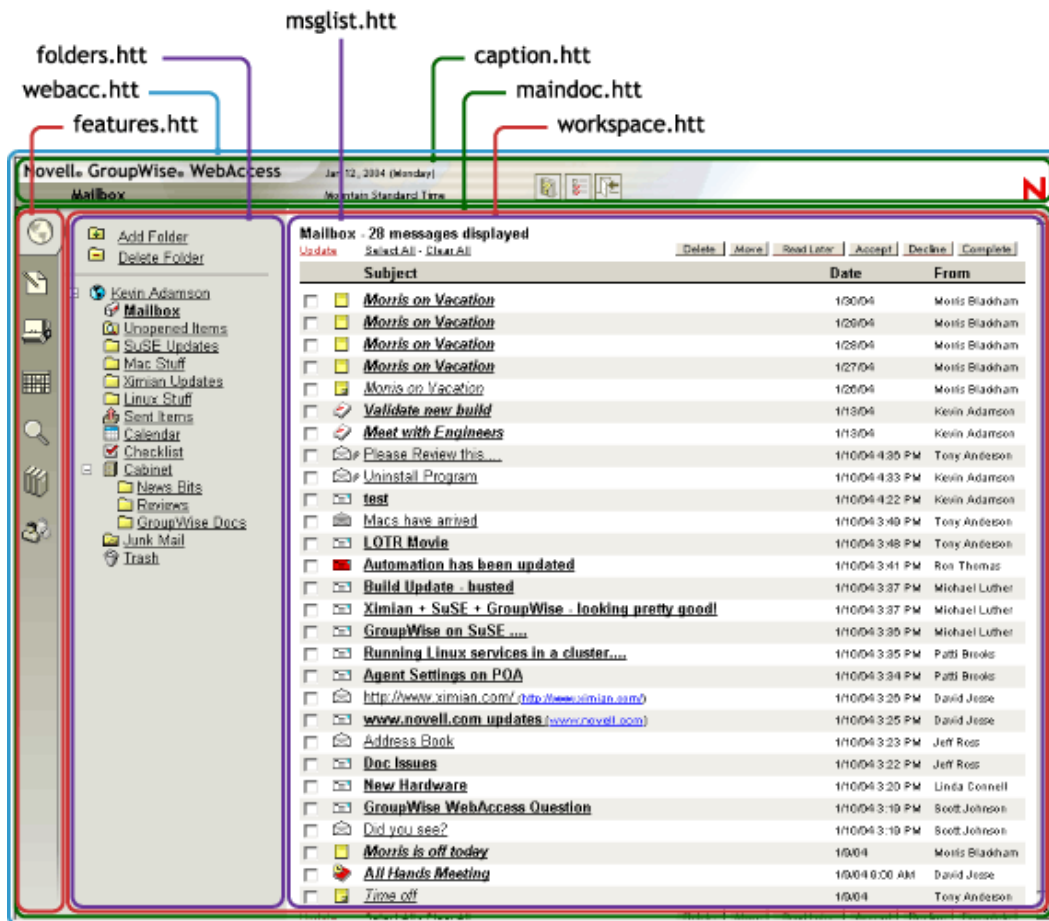
The Main Message Window is the dialog that appears after you login.

For more information about the Mail Message Window, see

- ♦ Section 3.3.1, “Mail Message Window,” on page 28

The base file is webacc.htt, which includes caption.htt and maindoc.htt. The maindoc.htt includes the features.htt and workspace.htt. Note that the interface for features.htt is showing that you are in the message view reading messages. The workspace.htt file contains folders.htt and msglist.htt. It is also used to toggle between other features, such as Document Management.

Figure 3-2 Main Message Window



3.3.1 Mail Message Window

The Mail Message Window appears when the user double-clicks on one of the incoming or outgoing messages. It includes the `captbar.inc` file.

Figure 3-3 Mail Message Window



3.4 Modifying the Compose Windows

When the paper and pencil icon is selected, a separate window appears with the contents of send.htm. The send.htm file contains the speller.applet when Java-enhance mode is turned on. The spell check allows you to spell check the message you are composing to send. The send.htm file also contains the captbar.inc file that you can modify to change the logo on this screen and any other like screens that pop up when selected.

Figure 3-4 Compose Window

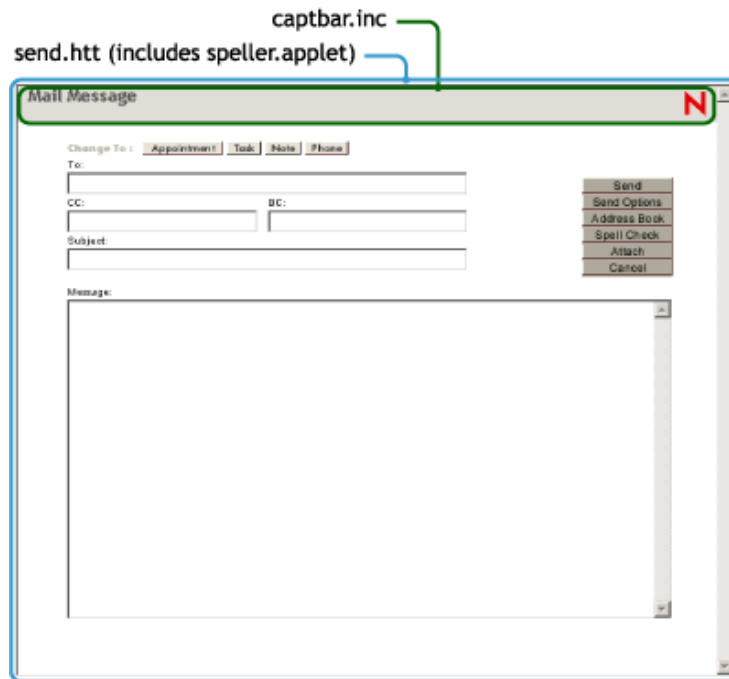
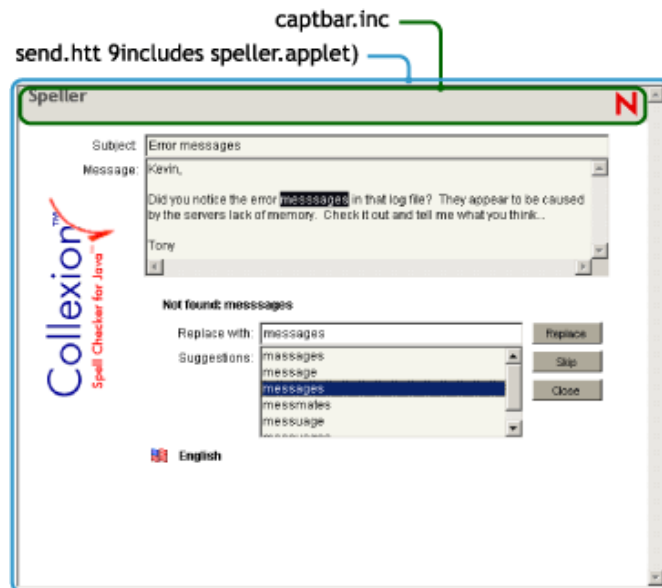


Figure 3-5 Speller Applet



3.5 Modifying the Address Book Windows

The Address Book is available in both Java-enhanced mode and HTML mode. They appear as follows.

Figure 3-6 Address Book Window, Java-Enhanced Mode

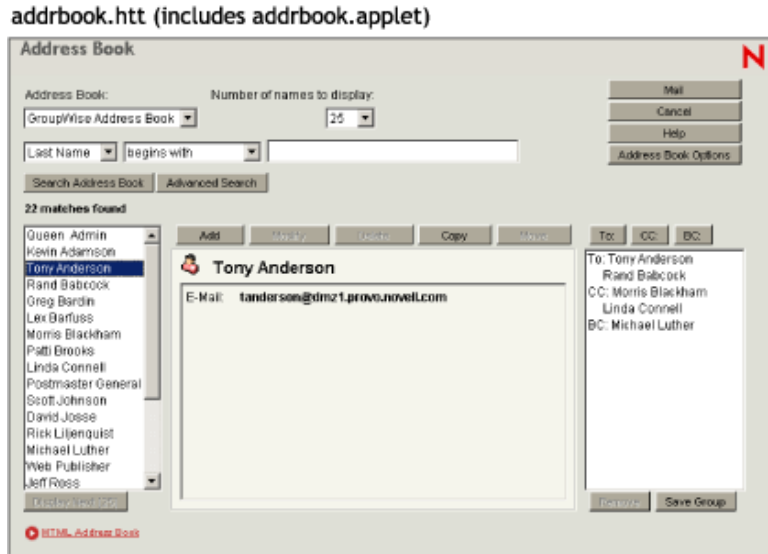
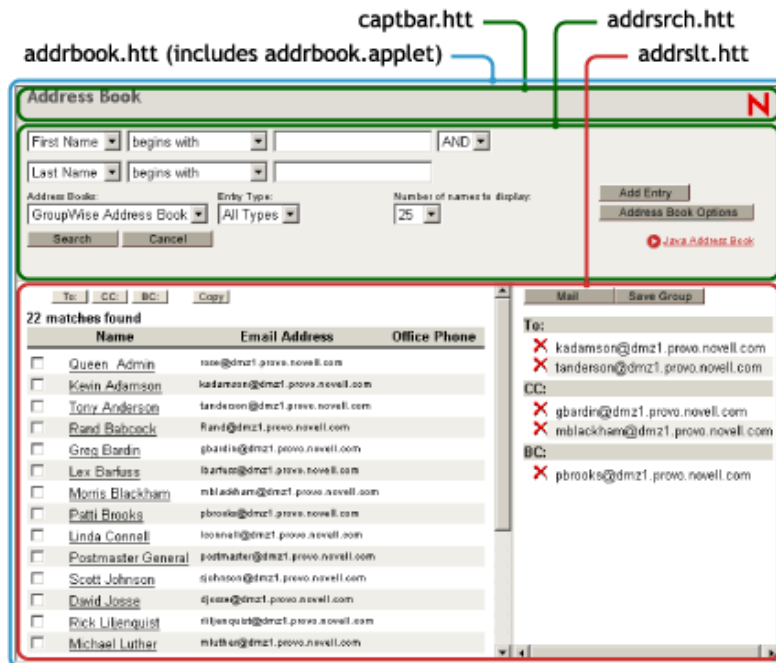


Figure 3-7 Address Book Window, HTML Mode



3.6 Modifying the Calendar Windows

The calendar view displays three different views in both Java-enhanced and HTML views:

- ◆ Section 3.6.1, “Daily Calendar,” on page 31
- ◆ Section 3.6.2, “Weekly Calendar,” on page 33
- ◆ Section 3.6.3, “Monthly Calendar,” on page 35

3.6.1 Daily Calendar

Views for the daily calendar include the following:

Figure 3-8 Daily Calendar, Java-Enhanced Mode

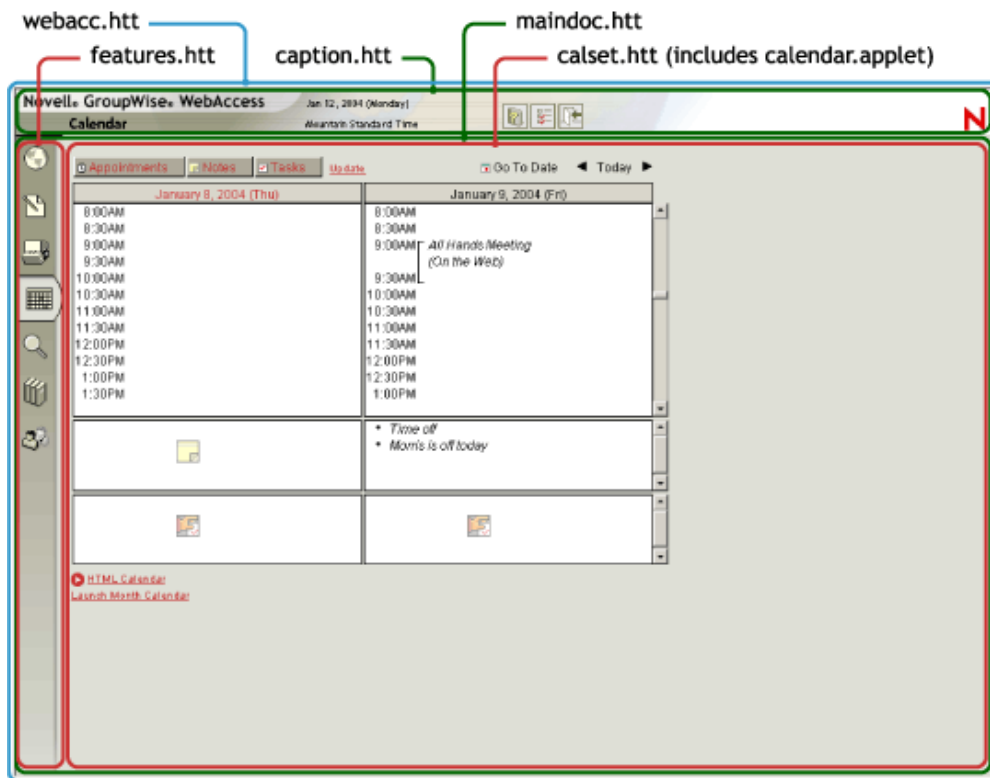
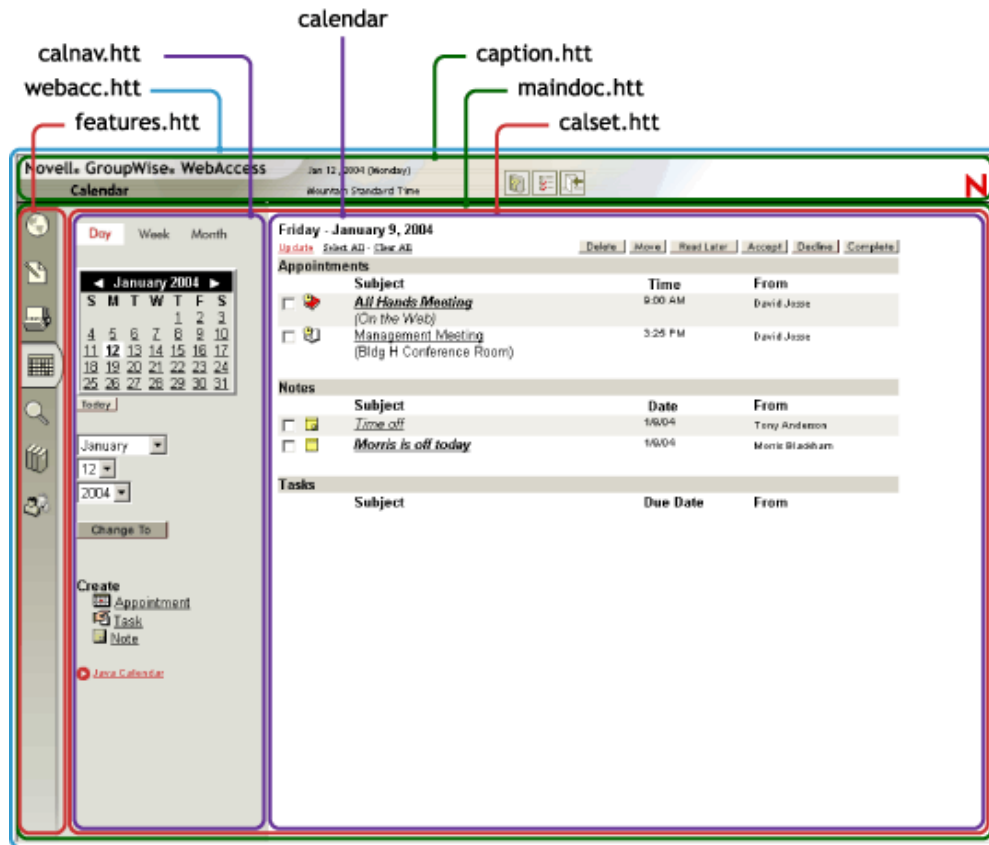


Figure 3-9 Daily Calendar, HTML Mode



3.6.2 Weekly Calendar

Views for the weekly calendar include the following:

Figure 3-10 Weekly Calendar, Java-Enhanced Mode

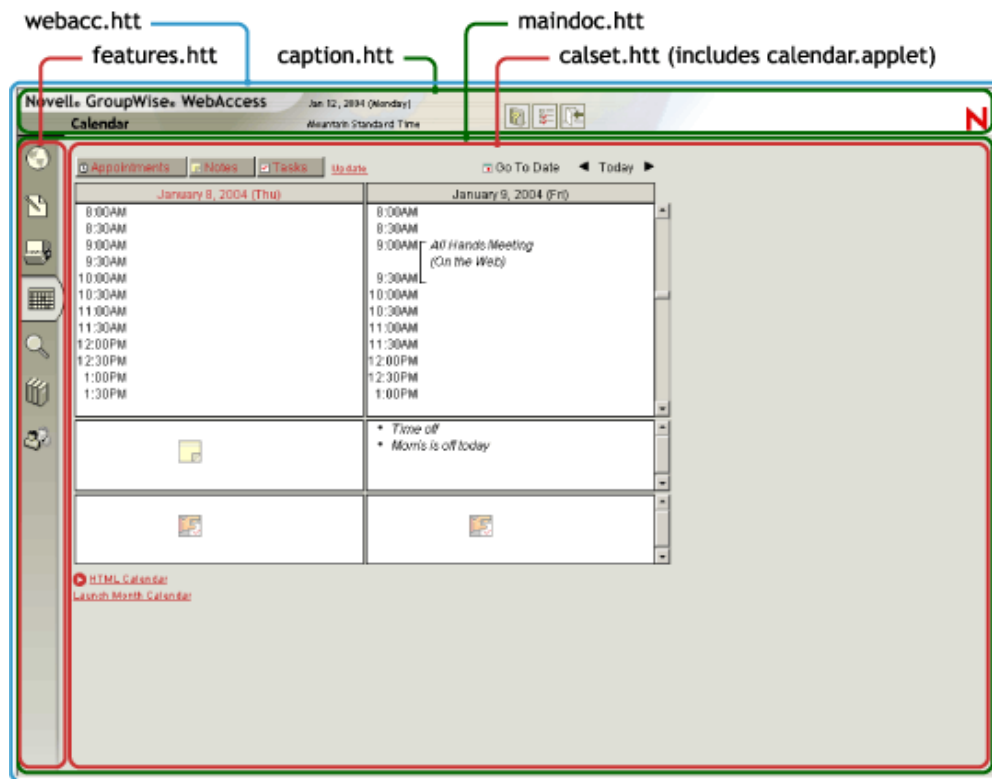
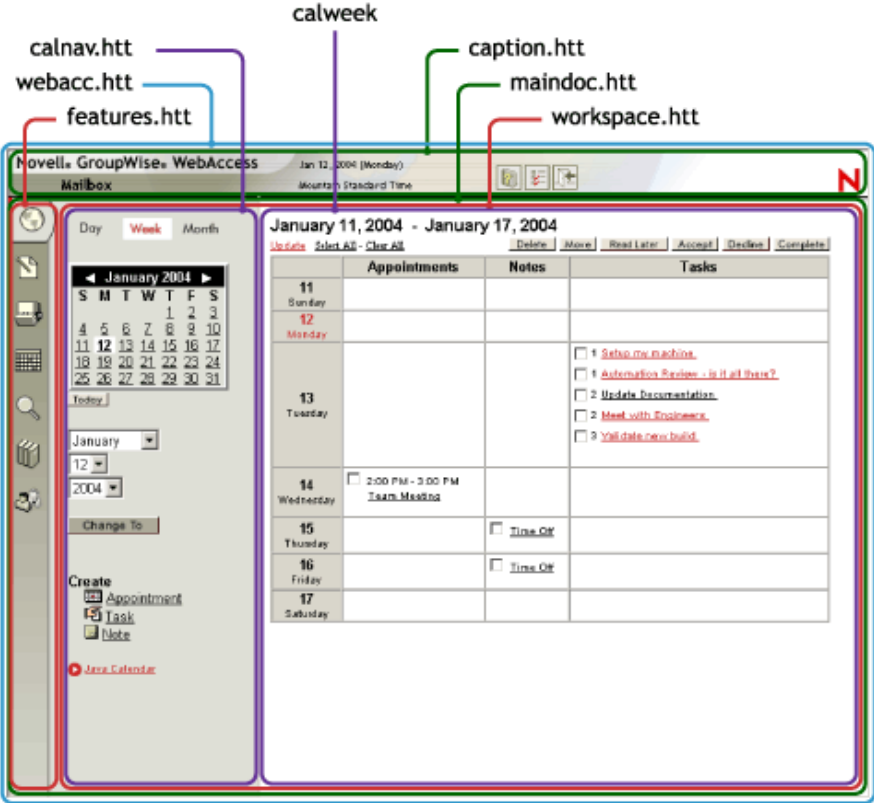


Figure 3-11 Weekly Calendar, HTML Mode



3.6.3 Monthly Calendar

Views for the monthly calendar include the following:

Figure 3-12 Monthly Calendar, Java-Enhanced Mode

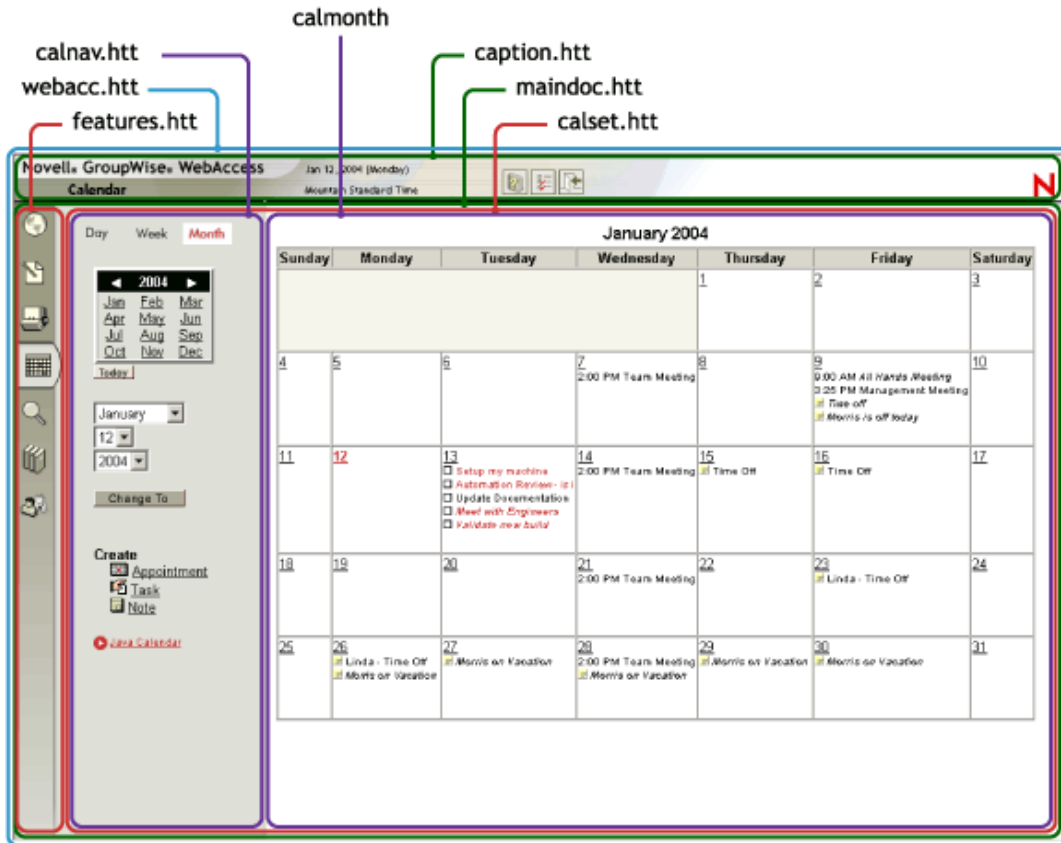
The screenshot shows a Java applet window titled "GroupWise Month Calendar" displaying a monthly calendar for January 2004. The calendar is presented as a grid with columns for each day of the week (Sunday through Saturday). The days of the month are arranged in a standard calendar format, starting with Sunday, January 4th, and ending with Saturday, January 31st. The 12th of the month is highlighted in red. Various events and tasks are listed in the cells, including team meetings, time-off notifications for individuals like Linda and Morris, and task lists for specific days like the 13th. At the bottom of the calendar grid, there are navigation buttons for each month of the year (Jan through Dec), an "Update" button, and navigation arrows for "Today". The window title bar at the bottom indicates it is a "Java Applet Window".

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	2	3
4	5	6	7 2:00 PM Team Meeting	8	9 9:00 AM All Hands Meet 3:25 PM Management M ☑ Time off ☑ Morris is off today	10
11	12	13 ☐ Setup my machine ☐ Automation Review - ☐ Update Documentation ☐ Meet with Engineers	14 2:00 PM Team Meeting	15 ☑ Time Off	16 ☑ Time Off	17
18	19	20	21 2:00 PM Team Meeting	22	23 ☑ Linda - Time Off	24
25	26 ☑ Linda - Time Off ☑ Morris on Vacation	27 ☑ Morris on Vacation	28 2:00 PM Team Meeting ☑ Morris on Vacation	29 ☑ Morris on Vacation	30 ☑ Morris on Vacation	31

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Update << Today >>

Java Applet Window

Figure 3-13 Monthly Calendar, HTML Mode



3.7 Modifying the Document Management Windows

The Document Management Window is similar to the Main Message Window, except that workspace.htm calls two different files: liblist.htm and doclist.htm.

Figure 3-14 Documents

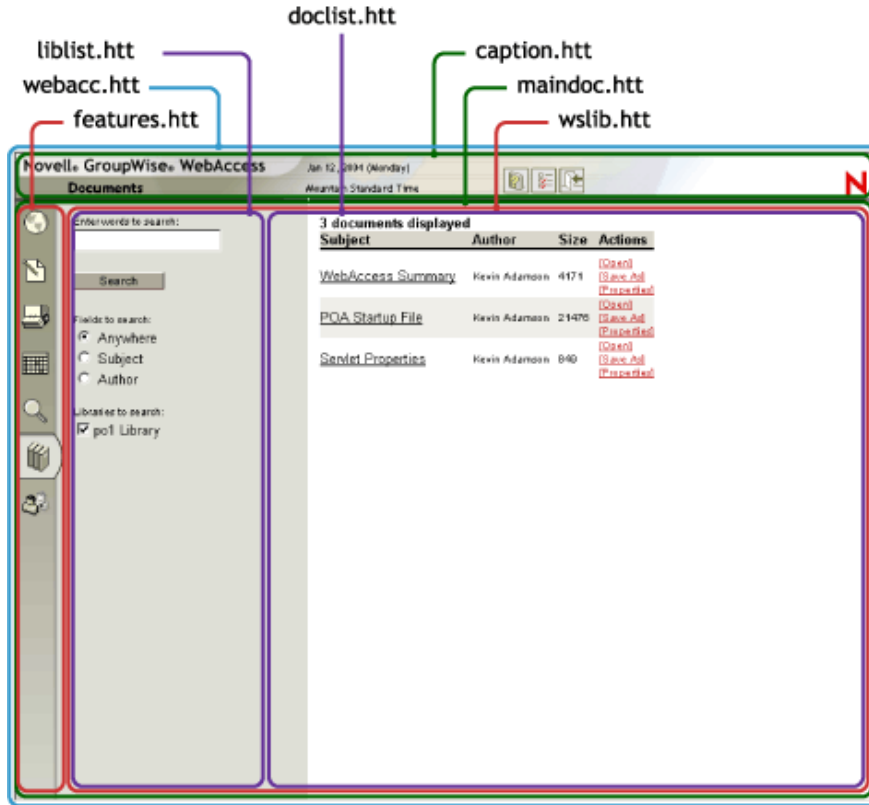
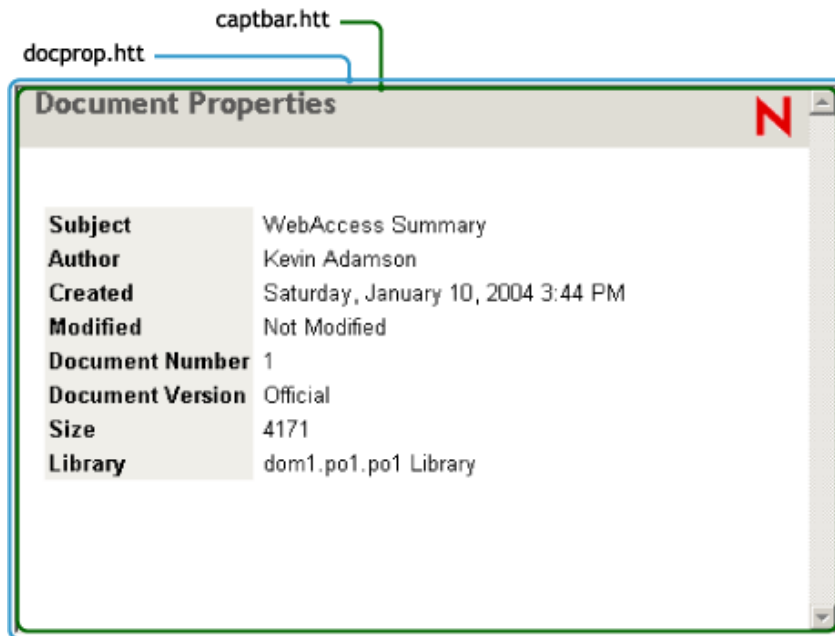


Figure 3-15 Document Properties



4 Template Keywords

This section contains the following case sensitive keywords:

- ♦ Section 4.1, “ABORT,” on page 39
- ♦ Section 4.2, “ALIAS,” on page 39
- ♦ Section 4.3, “COOKIE,” on page 41
- ♦ Section 4.4, “DEC,” on page 41
- ♦ Section 4.5, “DUMPALL,” on page 41
- ♦ Section 4.6, “HEADER,” on page 42
- ♦ Section 4.7, “IF...ELSEIF...ELSE...ENDIF IFDEF...ELSEIFDEF,” on page 42
- ♦ Section 4.8, “IGNOREWHITE and NOIGNOREWHITE,” on page 43
- ♦ Section 4.9, “INC,” on page 43
- ♦ Section 4.10, “INCLUDE,” on page 43
- ♦ Section 4.11, “NOOUTPUT,” on page 44
- ♦ Section 4.12, “OUTPUT,” on page 44
- ♦ Section 4.13, “SET,” on page 44
- ♦ Section 4.14, “SETGLOBAL,” on page 44
- ♦ Section 4.15, “STRING,” on page 45
- ♦ Section 4.16, “STRINGTABLE,” on page 45
- ♦ Section 4.17, “VAR,” on page 45
- ♦ Section 4.18, “WHILE...BREAK...CONTINUE...ENDWHILE,” on page 46

4.1 ABORT

Causes the processing of the template to abort. None of the template after the ABORT keyword is processed.

4.2 ALIAS

Provides a way to create constant value that can be used later in the template. When the alias name is encountered later in the template, it will be expanded to the constant value assigned to it. Using ALIAS is more efficient than using template variables created with the SET keyword because ALIAS

names are expanded when the template is initially compiled rather than when it is processed during a request. ALIAS essentially provides a shortcut notation within templates. When an ALIAS name is used, it must be between squiggly braces {} or it will not be expanded.

Example:

```
{ALIAS ITEMS_PER_PAGE 10}
{ALIAS ITEMS_REMAINING {ITEMS_PER_PAGE} - count}
{SET count = 1}
{WHILE count <= {ITEMS_PER_PAGE}}
  {SET remaining = {ITEMS_REMAINING}}
  This is item number {VAR count}. There are {VAR remaining} remaining.<BR>
  {INC count}
{ENDWHILE}
```

The preceding template produces the following output:

```
This is item number 1. There are 9 remaining.
This is item number 2. There are 8 remaining.
This is item number 3. There are 7 remaining.
This is item number 4. There are 6 remaining.
This is item number 5. There are 5 remaining.
This is item number 6. There are 4 remaining.
This is item number 7. There are 3 remaining.
This is item number 8. There are 2 remaining.
This is item number 9. There are 1 remaining.
This is item number 10. There are 0 remaining.
```

Note that if we had attempted to use {VAR {ITEMS_REMAINING}}, we would have encountered a compile error similar to the following:

```
Compile Error: aliastest.htt: Line 45: Invalid name for a variable: 10. Cannot load
file: aliastest.htt.
```

The reference to {VAR {ITEMS_REMAINING}} does not work because {ITEMS_REMAINING} is not a valid variable name. It expands to {ITEMS_PER_PAGE} - count, ITEMS_PER_PAGE expands to 10, and the result is 10 - count. Because the VAR keyword expects the name of a variable, and 10 is not a valid variable name, the error occurs.

Sometimes template data is provided in the form of numbers, making templates difficult to read and maintain. In this situation, you can use the ALIAS keyword to make your templates more readable. For example, suppose a variable named Object.type is returned, and its possible values are numeric:

```
{ALIAS TYPE_USER 1}
{ALIAS TYPE_GROUP 2}
.
.
.
{IF Object.type == {TYPE_USER}}
  {! Show information about the user object}
  {TYPE_USER} is a user object.
{ELSEIF Object.type == {TYPE_GROUP}}
  {! Show information about the group object}
  {TYPE_GROUP} is a group object.
{ENDIF}
```


4.3 COOKIE

Sends a cookie to the web browser. The `COOKIE` keyword requires the parameters described in the following table, in the order listed:

Parameter	Description
Name	The name of the cookie.
Value	The value of the cookie.
Domain	The domain to which the browser will respond with the cookie.
URL	The URL to which the browser will respond with the cookie.
MaxAge	The maximum age of the cookie in seconds. After this number of seconds has elapsed, the cookie expires. Zero deletes the cookie. A negative number or blank value indicates the cookie is not persistent and will be deleted when the browser exits.
Secure	Values of true, yes, or 1 indicate that the cookie should not be sent unless a secure protocol, such as HTTPS, is used. Evaluation of the value of this parameter is not case-sensitive.

You must supply a non-empty value for the Name and Value parameters, but you can omit values for the other parameters. Empty values must be represented with double quote characters.

Because headers and cookies must be sent before any data, the `COOKIE` keyword must precede any data in the template, including spaces, tabs, carriage returns, and line feeds. The easiest way to ensure that no data is sent before the `COOKIE` keyword is processed is to include the `NOOUTPUT` keyword as the very first item in the template, followed by the `COOKIE` keyword and the `OUTPUT` keyword, as shown in the following example:

```
{NOOUTPUT}
{COOKIE "user-id" "jsmith" "" "" "" ""}
{OUTPUT}
. . .
```

For more information about the `NOOUTPUT` and `OUTPUT` keywords, see [Section 4.11, “NOOUTPUT,” on page 44](#) and [Section 4.12, “OUTPUT,” on page 44](#).

4.4 DEC

Decrements the value of a variable. `DEC` can only be used with numeric variables.

4.5 DUMPALL

Causes all variable names and values to be inserted into the document. It allows you to generate a document that lists all the variables and strings that are defined and accessible to the template.

4.6 HEADER

Sends a header to the browser. The HEADER keyword requires two parameters: the name of the header, and the value of the header.

Because headers and cookies must be sent before any data, the HEADER keyword must precede any data in the template, including spaces, tabs, carriage returns, and line feeds. The easiest way to ensure that no data is sent before the HEADER keyword is processed is to include the NOOUTPUT keyword as the very first item in the template, followed by the HEADER keyword and the OUTPUT keyword, as shown in the following example:

```
{NOOUTPUT}
{HEADER "pragma" "no-cache"}
{OUTPUT}
. . .
```

For more information about the NOOUTPUT and OUTPUT keywords, see [“NOOUTPUT” on page 44](#) and [“OUTPUT” on page 44](#).

4.7 IF...ELSEIF...ELSE...ENDIF IFDEF...ELSEIFDEF

Allows conditional inclusion of text into the document. The IF keyword requires a conditional expression. If the conditional expression evaluates to true, the section in the template that follows the IF keyword will be processed until an ELSE, ELSEIF, or an ENDIF is encountered.

If the conditional expression evaluates to false, processing in the template will skip to the next ELSE, ELSEIF, or ENDIF.

Conditional expressions can be combined with && and || for AND and OR expressions respectively.

IFDEF and ELSEIFDEF can be used in place of IF and ELSEIF. IFDEF and ELSEIFDEF require the name of a variable. If the variable exists, the condition evaluates to true.

You can also test the existence of a variable with DEF and NOTDEF. For example,

```
{IF a == b || (DEF c && NOTDEF d)}
{ENDIF}
```

Either a equals b, OR c is defined AND d is not defined

There must be a matching ENDIF keyword for every IF keyword.

```
{IFDEF Folder.count && Folder.count > 0}
{IF Folder.type == Folder.UNIVERSAL}
<IMG src="universal.gif">
{ELSEIF Folder.type == Folder.SHARED}
<IMG src="shared.gif">
{ELSE}
<IMG src="folder.gif">
{ENDIF}
{ELSE}
There are no folders.
{ENDIF}
```

4.8 IGNOREWHITE and NOIGNOREWHITE

Causes multiple white space characters (space, tab, new lines) that occur in succession in the template to be collapsed to a single white space character. It reduces the size of the document being generated.

NOIGNOREWHITE undoes the effect of IGNOREWHITE (turns off IGNOREWHITE).

IGNOREWHITE and NOIGNOREWHITE only affect the file in which they appear.

4.9 INC

Increments the value of a variable. INC can only be used with numeric variables.

4.10 INCLUDE

Causes the contents of another template file to be included into the current template at the point where the INCLUDE keyword is encountered. The name of the file to be included should not be a variable name and it should not be inside quote characters. Included files are compiled in as part of the file in which they are included. All variables are shared between the including file and the file it includes.

```
{! setfocus.inc }
<SCRIPT LANGUAGE="JavaScript">
<!--
function setFocus()
{
document.forms[0].elements[0].focus();
}
//-->
</SCRIPT>
<HEAD>
<TITLE>My Page</TITLE>
{INCLUDE setfocus.inc}
</HEAD>
```

This example generates the following HTML text:

```
<HEAD>
<TITLE>My Page</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function setFocus()
{document.forms[0].elements[0].focus();}
//-->
</SCRIPT>
</HEAD>
```

4.11 NOOUTPUT

Prevents any data from being sent to a web browser, including white space, spaces, tabs, and line breaks. This is useful for situations when you need to use the COOKIE or HEADER keywords, because those keywords must be processed before any template data is sent. Use the OUTPUT keyword to enable the sending of template data to the web browser.

The following template excerpt shows how to use the NOOUTPUT and OUTPUT keywords:

```
{NOOUTPUT}
{HEADER "pragma" "no-cache"}
{OUTPUT}
. . .
```

4.12 OUTPUT

Enables the sending of template data to a web browser after it has been disabled using the NOOUTPUT keyword.

4.13 SET

Assigns a value to a variable. If this is its first use, the variable will be created and assigned the specified value. If the variable already exists, its value will be changed to the value specified.

Requires a variable name, followed by an equals sign (=), followed by a value. The value can be a string, an arithmetic expression, or a string expression.

```
{SET a = 5}
{SET b = 3}
{SET c = a + b + 4}
{SET h = "Hello"}
{SET w = "World"}
{! string expression }
{SET hw = {VAR h}{VAR w}}
{! Concatenation of a string and variable}
{SET mornin = "Good morning "{VAR w}} {! No space between closing quote and {VAR
w} }
{VAR hw}<BR>
c = {VAR c} <BR>
{VAR mornin}
```

This example produces the following output:

```
Hello World
c = 12
Good morning World
```

4.14 SETGLOBAL

Works the same as SET except that the variable set is saved as a global variable so it can be accessed on later requests and in other templates.

4.15 STRING

The template refers to values in the String table by using the `STRING` keyword followed by the name whose value is desired. The value will replace the `STRING` keyword.

```
#strings_sp.properties
btnYes=Si
btnNo=No

{! Confirm.inc }
{STRINGTABLE strings}
<FORM Method=put Action="...">
  <INPUT Type="submit" Name="Yes" Value="{STRING btnYes}">
  <INPUT Type="submit" Name="No" Value="{STRING btnNo}">
</FORM>
```

This example would generate the following HTML text:

```
<FORM Method=put Action="...">
  <INPUT Type="submit" Name="Yes" Value="Si">
  <INPUT Type="submit" Name="No" Value="No">
</FORM>
```

4.16 STRINGTABLE

Loads a file known as a String Table. This has the format of a Java properties file which contains name value pairs in the form

```
Name1=Value1
Name2=Value2
```

Lines starting with the pound sign (#) are considered comments and are ignored. Blank lines are also ignored.

`STRINGTABLE` requires the name of the string table to load. An underscore character followed by the current language is appended to the to the file name. For example, if the template contains `{STRINGTABLE strings}`, the file that would be loaded if the current language were English (en) would be `strings_en.properties`. If this file is not found, `strings.properties` will be loaded.

Refer to the servlet's URL interface documentation to determine how the servlet sets the language.

If a string name appears in more that one include file, the value in the file specified by the last `STRINGTABLE` keyword will override any previous occurrences.

String tables can be used to facilitate localization of templates and to standardize commonly used text such as background colors, fonts, labels, etc.

4.17 VAR

Causes a variable to be inserted into the template. The variable could have been created with the `SET` or `SETGLOBAL` keyword. It could also be a value that a provider returned to the servlet (see providers), a name found in the HTTP request, or a value specified in the `webacc.cfg` file.

The `VAR` keyword takes one parameter (except when that parameter uses replaceable parameters):

- ◆ The name of the variable whose value is to be inserted into the template
- ◆ An arithmetic expression
- ◆ Function

If the parameter for the VAR keyword is a variable whose value is a string containing replaceable parameters, VAR can have additional parameters that are used as the values for the replaceable parameters:

```
{SET replaceable = "This is the {0} parameter, and this is the {1} parameter"}  
{VAR replaceable "first" "second"}
```

4.18 WHILE...BREAK...CONTINUE...ENDWHILE

Requires a conditional expression. If the conditional expression evaluates to true, the section of the template is processed. When the ENDWHILE is encountered, the expression is evaluated again. The section of the template is processed repeatedly until the expression evaluates to false.

If a BREAK keyword is encountered before the ENDWHILE, processing is discontinued until immediately after the ENDWHILE keyword.

If a CONTINUE keyword is encountered, processing moves back to the WHILE keyword for re-evaluation. Anything between the CONTINUE and ENDWHILE keywords are not processed.

```
{SET maxLines=10}  
{SET i = 0}  
{WHILE i < Folder.count}  
{IF i > maxLines}  
{BREAK}  
{ENDIF}  
{IF Folder[i].type == "FOLDER-TYPE-CALENDAR"}  
{INC i}  
{CONTINUE} {! Skip the calendar folder }  
{ENDIF}  
{VAR Folder[i].name}<BR>  
{INC i}  
{ENDWHILE}
```

5 Template Functions

WebAccess provides many built-in functions you can use to cause special processing to occur. A function cannot appear in a template alone, but must serve as a parameter of a keyword, part of an expression that is a parameter of a keyword, or a parameter of another function. Function names are case sensitive.

Functions usually appear within a *SET*, *STRING*, or *VAR* keyword. The following example uses the *substring* function as the parameter of the *VAR* keyword.

```
{SET TisATest="this is a test"}
{VAR substring(TisATest, 8, 6)}
{! output: "a test"}
```

You need to be aware of the following rules concerning functions:

- ◆ If a function is used in an expression, the expression must be enclosed in parentheses:

```
{SET tomorrow = (today() + 86400000)}
```

- ◆ Expressions used as function parameters must be enclosed in parentheses:

```
{SET curDay = today()}
Tomorrow is {VAR date((curDay + 86400000), {STRING fullDate})}
```

- ◆ Functions can serve as parameters to other functions:

```
{VAR date(today(), {STRING fullDate})}
```

- ◆ Variables and names of strings in string tables cannot have the name of a function. For more information about string tables see [Section 4.16, "STRINGTABLE," on page 45](#).

```
{SET date = today()}
{! Illegal. date is the name of a function. }
```

This document describes the following template functions and their syntax, parameters, and return values:

- ◆ ["addToDate" on page 49](#)
- ◆ ["createDate" on page 50](#)
- ◆ ["date" on page 51](#)
- ◆ ["daysInMonth" on page 53](#)
- ◆ ["endsWith" on page 54](#)
- ◆ ["find" on page 55](#)
- ◆ ["findLast" on page 56](#)
- ◆ ["firstWeekDayOfMonth" on page 57](#)
- ◆ ["firstDayOfWeek" on page 58](#)
- ◆ ["formatMessage" on page 59](#)
- ◆ ["getHeader" on page 60](#)

- ◆ “length” on page 61
- ◆ “linkEncodedURLs” on page 62
- ◆ “linkURLs” on page 63
- ◆ “listToArray” on page 65
- ◆ “listToSortedArray” on page 66
- ◆ “noScript” on page 67
- ◆ “replaceWith” on page 68
- ◆ “startsWith” on page 69
- ◆ “substring” on page 70
- ◆ “today” on page 71
- ◆ “todayTZ” on page 72
- ◆ “toDisplay” on page 73
- ◆ “toHTML” on page 74
- ◆ “toHTMLExt” on page 75
- ◆ “toLowerCase” on page 76
- ◆ “toQuote” on page 77
- ◆ “toScript” on page 78
- ◆ “toSpellerText” on page 79
- ◆ “toUpperCase” on page 80
- ◆ “toWMLExt” on page 81
- ◆ “toXML” on page 82
- ◆ “trim” on page 83
- ◆ “urlEncode” on page 84
- ◆ “xmlDecode” on page 85
- ◆ “xmlEncode” on page 86

addToDate

Adds time to the date.

Syntax

```
addToDate(date, type, amount)
```

Parameters

date

Specifies the date value as a 64-bit number representing the number of milliseconds since January 1, 1970.

type

Type of time units added to date. Possible values are SECONDS, MINUTES, HOURS, DAYS, WEEKS, MONTHS, YEARS.

amount

Number of time units to add to date.

Returns

Returns a numeric value representing the specified date.

createDate

Creates a date variable representing the date of the month, day and year passed in.

Syntax

```
createDate(month, day, year)
```

Parameters

month

Specifies the month: 0=January.

day

Specifies the day of the month: 1=Day one.

year

Specifies the year with a four-digit number: 1999 or 2000.

Returns

Returns a numeric value representing the specified date.

date

Converts a date into a human readable string based on the specified format.

Syntax

```
date(date, format)
```

Parameters

date

Specifies the date value as a 64-bit number representing the number of milliseconds since January 1, 1970.

format

Specifies the format to be used to create the human readable text version of the string.

Returns

Returns the formatted date string.

Remarks

To add an hour to a time, add 3,600,000 to the date value. To add a day to a time, add 86,400,000 to the date value.

The following are the date formats:

Symbol	Meaning	Presentation	Example
G	era designator	text	AD
y	year	numeric	1998 (yyyy) or 98 (yy)
M	month	text and numeric	July (MMMM) or 07 (MM)
d	day	numeric	10
h	hour in a.m./p.m. (1-12)	numeric	12
H	hour in day (0-23)	numeric	0
m	minute	numeric	30
s	second	numeric	55
S	millisecond	numeric	457
E	day in week	text	Tuesday
D	day in year	numeric	189
F	day of week in month	numeric	2 (second Wednesday in July)
w	week in year	numeric	27
W	week in month	numeric	2

Symbol	Meaning	Presentation	Example
a	a.m./p.m.	text	p.m.
k	hour in day (1-24)	numeric	24
K	hour in a.m./p.m.	numeric	0
z	time zone	text	Pacific Standard Time
>	escape for text	delimiter	
>=	single quote	literal	>

Examples:

```
{! 10/27/99 }
Today is {VAR date(Calendar.today, "M/d/yy")}
```

```
{! 2:30 p.m. }
Meeting is at {VAR date(Message.startTime, "h:mm a")}
```

```
{! Friday, November 13, 1999}
Today is {VAR date(Calendar.today, "EEEE, MMMM d, yyyy")}
```

daysInMonth

Calculates the number of days in a month.

Syntax

```
daysInMonth (date)
```

Parameters

date

Specifies a numeric value representing the date.

Returns

Returns the number of days in the month of the date that is passed in.

endsWith

Determines if a string ends with the specified string pattern.

Syntax

```
endsWith(string, endPattern)
```

Parameters

string

Specifies the case-sensitive string to be checked.

endPattern

Specifies the case-sensitive pattern to look for at the end of string.

Returns

Returns 1 if the string ends with endPattern.

find

Finds the first occurrence of srchstring in string and returns its position.

Syntax

```
find(string, srchstring)
```

Parameters

string

Specifies the original string.

srchstring

Specifies the string to search for.

Returns

Returns the position of srchstring in the string or -1 if not found.

findLast

Finds the last occurrence of srchstring in string and returns its position.

Syntax

```
findLast(string, srchstring)
```

Parameters

string

Specifies the original string.

srchstring

Specifies the string to search for.

Returns

Returns the position of srchstring in the string or -1 if not found.

firstWeekDayOfMonth

Calculates the day of the week that the first day of the month falls on.

Syntax

```
firstWeekDayOfMonth(date)
```

Parameters

date

Specifies the date variable.

Returns

Returns a value representing the first day of the month of the date that is passed in: 1 = Sunday, 2 = Monday . . . 7 = Saturday.

firstDayOfWeek

Calculates the first day of the week containing the given date.

Syntax

```
firstDayOfWeek (date, weekStart)
```

Parameters

date

Specifies the date variable.

weekStart

Specifies a number representing the day used as the first day of the week: 1 = Sunday, 2 = Monday . . . 7 = Saturday.

Returns

Returns a date value representing the day that begins the week containing the date passed in.

formatMessage

Formats a string using the passed in parameters.

Syntax

```
formatMessage(formatString, parameter0, parameter1, parameterX,  
numberOfParameters)
```

Parameters

formatString

Specifies the string to be formatted. This string contains a variable number of replaceable parameters (for example, {0}, {1}, etc.,).

parameter0

Specifies the string to be merged into the formatString where the {0} replaceable parameter is specified.

parameter1

Specifies the string to be merged into the formatString where the {1} replaceable parameter is specified.

parameterX

Specifies the string to be merged into the formatString where the {X} replaceable parameter is specified.

numberOfParameters

Specifies how many replaceable parameters are being passed into the function.

Returns

Returns a string that has been merged with the passed in replaceable parameters.

Remarks

formatMessage takes a variable number of parameters and provides a means to produce concatenated messages in a language-neutral way.

getHeader

Returns the value of an HTTP header.

Syntax

```
getHeader (header)
```

Parameters

header

Specifies the name of a header.

Returns

Returns the value of the header.

length

Returns the number of characters in string.

Syntax

```
length(string)
```

Parameters

string

Specifies the string whose length is to be returned.

Returns

Returns the number of characters in the string.

linkEncodedURLs

Creates an anchor tag around URLs within a string and URL encodes the link text.

Syntax

```
linkEncodedURLs(string, protoString, format, linksOnly, separator)
```

Parameters

string

Specifies the string in which to search for potential URLs.

protoString

Is reserved.

format

Specifies the format string: {0} refers to the actual link and {1} refers to the non-linked version of the URL.

linksOnly

Specifies whether to return a string containing only the formatted links, or a string containing the original text with the formatted links embedded where they occurred in the original text:

true Returns a string that only has the links found in the string parameter. The links will be formatted as per the format parameter, and will be separated by the string specified with the separator parameter.

false Returns a string containing the full contents of the string parameter with the formatted links embedded in it as they occurred in the original string.

separator

Specifies the string used to separate links when the linksOnly parameter is set to true (false means the separator is ignored.)

Remarks

The linkEncodedURLs method works just like [linkURLs \(page 63\)](#), except that the resulting link is URL encoded.

linkURLs

Creates an anchor tag around URLs within a string.

Syntax

```
linkURLs(string, protoString, format, linksOnly, separator)
```

Parameters

string

Specifies the string in which to search for potential URLs.

protoString

Is reserved.

format

Specifies the format string: {0} refers to the actual link and {1} refers to the non-linked version of the URL.

linksOnly

Specifies whether to return a string containing only the formatted links, or a string containing the original text with the formatted links embedded where they occurred in the original text:

true Returns a string that only has the links found in the string parameter. The links will be formatted as per the format parameter, and will be separated by the string specified with the separator parameter.

false Returns a string containing the full contents of the string parameter with the formatted links embedded in it as they occurred in the original string.

separator

Specifies the string used to separate links when the linksOnly parameter is set to true (false means the separator is ignored.)

Remarks

LinkURLs is useful for generating an HTML link in text that has a URL. For example, if string contains ? and Novell's home page is <http://www.novell.com>?, an anchor can be created around <http://www.novell.com> so that the end user can click on it to browse that page.

Assuming Message.message is: Visit <http://www.novell.com>. And the template contains these lines:

```
{SET fmtMsg=toHTML(Message.message)}:{VAR linkURLs(fmtMsg, "", "<A href={0}>{1}</A>" false, "")}
```

The generated output will be:

```
Visit <A href=http://www.novell.com>http://www.novell.com</A>
```

Another example is to assume Message.message is: Visit <http://www.novell.com> and <http://developer.novell.com>. And the template contains these lines:

```
{SET fmtMsg=toHTML(Message.message)}:{VAR linkURLs(fmtMsg, "", "<A href={0}>{1}</A>" true, " ")}
```

The generated output will be:

```
Visit http://www.novell.com and developer.novell.com (<A href=http://www.novell.com>http://www.novell.com</A>) (<A href=http://developer.novell.com>http://developer.novell.com</A>)
```


listToArray

Creates an array containing items from a delimited string.

Syntax

```
listToArray(textList, delimiters, arrayName)
```

Parameters

textList

Specifies a string containing items separated by any of the delimiters specified in the delimiters parameter.

delimiters

Specifies a string in which each character is a delimiter that can serve to separate the elements specified in the textList parameter.

arrayName

Specifies the name of the array that will be created.

Returns

Returns an empty string.

Remarks

Each item in textList will be created as a variable whose name begins with the name specified by arrayName, followed by a period, followed by a the index for that item. A final variable containing the number of items in the list will be created by using arrayName, followed by .count.

For example,

```
{VAR listToArray("one, two three;four,;seven", ", ;", "Numbers")}
```

will generate an array called Numbers that has the following values:

```
Numbers [1]=one  
Numbers [2]=two  
Numbers [3]=three  
Numbers [4]=four  
Numbers [5]=seven  
Numbers.count=5
```

For more information about using arrays, see [“Arrays” on page 20](#).

listToSortedArray

Creates an alphabetically sorted array of items from a delimited string.

Syntax

```
listToSortedArray(textList, delimiters, arrayName, ascend)
```

Parameters

textList

Specifies a string containing items separated by any of the delimiters specified in the delimiters parameter.

delimiters

Specifies a string in which each character is a delimiter that can serve to separate the elements specified in the textList parameter.

arrayName

Specifies the name of the array that will be created.

ascend

Specifies whether to sort the returned array in ascending or descending order.

- 1: Sort the array in ascending order.
- 2: Sort the array in descending order.

Returns

Returns an empty string.

Remarks

The listToSortedArray function works the same way as the listToArray function, except that the items in the array are sorted alphabetically. For example,

```
{VAR listToSortedArray("one, two three;four;;seven", ";", "Numbers", 1)}
```

will generate an array called Number that has the following values:

```
Numbers [1]=four  
Numbers [2]=one  
Numbers [3]=seven  
Numbers [4]=three  
Numbers [5]=two  
Numbers.count=5
```

For more information about using arrays, see [Section 2.7, "Arrays,"](#) on page 20.

noScript

Disables scripts in a string by converting tags to comments. It is used to send clean data back to the browser.

Syntax

```
noScript(string)
```

Parameters

string

Specifies the string to be cleaned. The original string is not modified.

Returns

Returns a string that has script tags converted to comments.

Remarks

noScript converts the following tags:

Tag	Result
<SCRIPT>	<!-- PT>
</SCRIPT>	</SCR -->
<EMBED>	<!-- D>
</EMBED>	</EM -->
<OBJECT>	<!-- CT>
</OBJECT>	</OBJ -->
<IFRAME>	<!-- AME>
</IFRAME>	</IFR -->
<FRAMESET>	<!-- ESET>
</FRAMESET>	</FRAME -->

replaceWith

Searches srchString, replacing occurrences of findString with replaceString.

Syntax

```
replaceWith(srchString, findString, replaceString)
```

Parameters

srchString

Specifies the value of this variable is not modified.

findString

Specifies the string to search for that will be replaced.

replaceString

Specifies the string that will replace each occurrence of findString.

Returns

Returns the string in which occurrences of findString have been replaced with replaceString.

startsWith

Determines if a string ends with the specified string pattern.

Syntax

```
startsWith(string, beginPattern)
```

Parameters

string

Specifies the string to be checked.

beginPattern

Specifies the pattern to look for at the beginning of string.

Returns

Returns 1 if string begins with startPattern.

substring

Extracts a substring from another string.

Syntax

```
substring(string, start, len)
```

Parameters

string

Specifies the original string.

start

Specifies the position in the string where the substring starts.

len

Specifies the length of the desired substring.

Returns

Returns a string of len characters beginning at index specified by start.

today

Returns the current Greenwich Mean Time (GMT) date and time.

Syntax

```
today
```

Returns

Returns a date variable representing the current GMT date and time. Use [date \(page 51\)](#) to display the date and time in the desired format.

todayTZ

Returns the current date and time using the time zone of the server where the servlet is running.

Syntax

```
todayTZ
```

Returns

Returns a date variable representing the current date and time of the server. Use [date \(page 51\)](#) to display the date and time in the desired format.

toDisplay

Converts plain text, including extended characters, to HTML.

Syntax

```
toDisplay(string)
```

Parameters

string

Specifies the string to be converted (original string is not modified).

Returns

Returns a string that has special characters converted to their HTML encoded strings.

Remarks

The toDisplay function works just like [toHTML \(page 74\)](#), except that it also converts characters with ASCII values greater than 127. For example, decimal character 128 is converted to `€`.

toHTML

Converts plain text to HTML.

Syntax

```
toHTML(string)
```

Parameters

string

Specifies the string to be converted (original string is not modified).

Returns

Returns a string that has special characters converted to their HTML encoded strings.

Remarks

The following characters are converted as indicated in the following table:

Character	Result
<	<
>	>
&	&
\$	$
" (double quote)	"
Carriage Returns/Line Feeds	
Tabs	 (repeatedly until a tab stop is encountered; tab stops are every four spaces)
Spaces	
(Escape)	

To convert extended characters (those with ASCII values greater than 127), use [toHTMLExt \(page 75\)](#) or [toDisplay \(page 73\)](#).

toHTMLExt

Converts plain text, including extended characters, to HTML.

Syntax

```
toHTMLExt(string)
```

Parameters

string

Specifies the string to be converted (original string is not modified).

Returns

Returns a string that has special characters converted to their HTML encoded strings.

Remarks

The toHTMLExt function works just like [toHTML](#) (page 74), except that it also converts characters with ASCII values greater than 127. For example, decimal character 128 is converted to `€`.

toLowerCase

Returns the string converted to lower case.

Syntax

```
toLowerCase(string)
```

Parameters

string

Specifies the string to be converted (its value is not modified).

Returns

Returns the lowercase version of the string.

toQuote

Converts single and double-quote characters within the passed in string to their XML encoded strings.

Syntax

```
toQuote(string)
```

Parameters

string

Specifies the string to be converted. The original string is not modified.

Returns

Returns a string in which all single and double quote characters are converted to their XML encoded strings.

Remarks

The following characters are converted as indicated in the following table:

Character	Result
' (single quote)	'
" (double quote)	"

toScript

Escapes single quotes, double quotes, and backslashes in a string by preceding each with a backslash, allowing the string to be used more easily in scripting languages.

Syntax

```
toScript(string)
```

Parameters

string

Specifies the string to be converted (original string is not modified).

Returns

Returns a string in which the single quotes, double quotes, and backslashes are preceded with a backslash.

toSpellerText

Returns a string that can be used with the GroupWise WebAccess speller applet.

Syntax

```
toSpellerText (string, hardReturn)
```

Parameters

string

Specifies a string to be converted (its value of this variable is not modified).

hardReturn

Specifies a string which will be used to identify carriage returns and line feeds in the text.

Returns

Returns a string that the speller applet can use to do a spell check.

toUpper

Returns the string converted to upper case.

Syntax

```
toUpper(string)
```

Parameters

string

Specifies the string to be converted (its value is not modified).

Returns

Returns the uppercase version of the string.

toWMLExt

Converts plain text, including extended characters, into HTML that is WML friendly.

Syntax

```
toWMLExt(string)
```

Parameters

string

Specifies the string to be converted. The original string is not modified.

Returns

Returns a string in which special characters are converted to their HTML encoded strings.

Remarks

The `toWMLExt` method works like [toHTMLExt \(page 75\)](#), except that it replaces a single dollar sign (\$) with a double dollar sign (\$\$), which is useful with certain wireless devices to enable the text to display correctly.

toXML

Converts plain text to XML.

Syntax

```
toXML(string)
```

Parameters

string

Specifies the string to be converted (original string is not modified).

Returns

Returns a string that has special characters converted to their XML encoded strings.

Remarks

The following characters are converted as indicated in the following table:

Character	Result
<	<
>	>
&	&
' (single quote)	'
" (double quote)	"
(Escape)	

trim

Removes beginning and ending white space from string.

Syntax

```
trim(string)
```

Parameters

string

Specifies the string to be trimmed (its value is not modified).

Returns

Returns a new string that now has white space at the beginning or end.

urlencode

Encodes a string.

Syntax

```
urlencode(string)
```

Parameters

string

Specifies the string to be encoded (its value is not modified).

Returns

Returns the URL-encoded version of the string.

Remarks

urlencode is useful for passing data on a URL. Characters that are not valid in a URL are encoded. Spaces are replaced with '+', special characters are escaped with %nn.

xmlDecode

Converts XML encoded strings to plain text.

Syntax

```
xmlDecode(string)
```

Parameters

string

Specifies the string to be converted. The original string is not modified.

Returns

Returns a string without any XML encoded characters.

Remarks

Characters represented by a #nn sequence are converted to their plain text equivalent characters. For example, the string sequence of #39 is converted to a single quote character.

xmlEncode

Converts plain text to XML.

Syntax

```
xmlEncode(string)
```

Parameters

string

Specifies the string to be converted (original string is not modified).

Returns

Returns a string that has special characters converted to their XML encoded strings.

Remarks

The following characters are converted as indicated in the following table:

Character	Result
<	<
>	>
&	&
' (single quote)	'
" (double quote)	"
(Escape)	

6 Servlet Concepts

Novell WebAccess Servlet is a flexible framework to generate a web interface for an application. The Servlet architecture consists of the servlet, providers, and templates. The servlet receives requests from a browser or other HTTP client, and handles those requests by calling the appropriate provider. The provider does the requested work and returns the results. The results are combined with a template to produce a text-based document (HTML, XML, etc.), which is then sent back to the browser or HTTP client.

This section covers the following topics:

- ♦ [Section 6.1, “Configuration Settings,” on page 87](#)
- ♦ [Section 6.2, “Java Packages,” on page 92](#)
- ♦ [Section 6.3, “Providers,” on page 93](#)
- ♦ [Section 6.4, “Localization and Internationalization,” on page 94](#)

6.1 Configuration Settings

The WebAccess servlet is controlled by configuration settings and by the HTTP requests it receives (see [“Configuring the Servlet” on page 95](#)). Configuration settings are loaded when the servlet starts up and exist throughout the life of the servlet. HTTP requests received by the servlet cause the servlet to carry out those requests, often by calling a provider.

All settings in the configuration file are accessible as variables in the templates. Blank lines or lines that begin with # are ignored. Each of the settings in the configuration file consists of the name of a setting, followed by an equals sign (=), followed by the value for the setting, as follows:

Value	Meaning
Admin.WebConsole.enable	Specifies whether to allow the Administration Web Console tool to be available from a URL (<code>http://<server>/servlet/webacc?action=Admin.Open</code>): true or false.
Admin.WebConsole.username	Specifies the user name to use when logging on to the Administration Web Console.
Admin.WebConsole.password	Specifies the password to use when logging on to the Administration Web Console.
Application.Display.version	Specifies the product version or name of the GroupWise WebAccess Application. (For example, 8.0.)
Application.Module.version	Specifies the specific service pack version of the GroupWise WebAccess Application. (For example, 8.0.3.)
Application.Copyright.year	Specifies the copyright year of the GroupWise WebAccess Application.

Value	Meaning
Document.View.excludeDocExtensions	Specifies the view option is not allowed, only for the specified document types. To allow the view option for all document extensions, do not set this option. Extensions are comma delimited with no white space between entries. (For example, txt,doc,xls,ppt.)
Document.View.maxSize	Specifies a maximum size limit for an attachment to be viewed. When reading an item with an attachment larger than the specified size, the view link for that attachment is not available.
Document.ViewNative.docExtensions	Specifies that the view native option is allowed only for the specified document types. To allow the view native option for all document extensions, do not set this option. Extensions are comma delimited with no white space between entries. (For example, txt,doc,xls, ppt.)
File.Upload.path	Specifies the name of the directory where attachments are uploaded they are added to messages. Attachments will be temporarily stored in the specified directory until the message has been successfully sent or canceled.
Help.url	Specifies the root of the URL for the HTML help files. The language directory and default.htm will be added to this URL to form the complete URL for the help file. For example, if com/novell/webaccess/help is specified, the final URL (assuming the language is English) will be com/novell/webaccess/help/en/default.htm.
Home.Speller.url	Specifies the URL used to invoke the Speller Servlet from the template files. This value is useful if the filter string (/servlet/) is changed to something other than the default.
Language.default	Specifies the default user interface language (as a two-character ISO language code).
Log.Console.enable	Specifies whether console logging is enabled (true = enabled).
Log.enable	Specifies whether file logging is enabled (true = enabled).
Log.language	Specifies the language of the log file (as a two-character ISO language code).
Log.level	Specifies the level of logging: normal verbose diagnostic
Log.maxAge	Specifies the number of days that log files should be kept. Files older than the number of days specified will be deleted.
Log.maxSize	Specifies the maximum accumulated size (in kilobytes) of all the files in the log directory. When the accumulated size exceeds the specified size, the oldest files will be deleted.
Logout.url	Specifies the URL of the page that should be presented when the user logs out of WebAccess. If not present, the login page will be returned.
Logout.Url.xxx	Specifies a logout URL for a specific interface. (For example, Logout.Url.frames=http://www.novell.com.) Each interface (such as simple, webclip, etc.,) can have its own logout URL.
Log.path	Specifies the path where log files are to be written.

Value	Meaning
Log.timeStamp	Specifies the date/time format to use when time stamping entries in the log file.
Provider.default	Specifies the name of the provider to use (when no provider is specified) in an HTTP request.
Provider.GWAP. BusySearch.timeOut	Specifies the number of minutes to allow a GWAP provider the busy search operation.
Provider.GWAP.class	Specifies that there is a provider named GWAP, and specifies the class file that implements this provider.
Provider.GWAP.Config.file	Specifies the path to the configuration file for the GWAP provider.
Provider.LDAP.class	Specifies that there is a provider named LDAP, and specifies the class file that implements this provider.
Provider.LDAP.Config.file	Specifies the path to the configuration file for the LDAP provider.
Provider.xxxx.class	Specifies that the name between Provider. and .class is a provider that can be specified in HTTP requests (see “Specifying a Provider” on page 96).
Security.Authenticate.header	Specifies what remote computers are trusted for receiving the authentication header. Multiple addresses can be on the line, separated by commas. (They can be any mixture of IP addresses or Domain names.)
Security.Authenticate.Header. start	Specifies the default template to use to merge the results of the login when using the authentication header instead of a login page form.
Security.HtmlDocument. disableScripts	Specifies whether WebAccess should disable potential security risks in HTML documents: true or false.
Security.HtmlAttachment. disableScripts	Specifies whether WebAccess should disable potential security risks in HTML attachments: true or false.
Security.Logout.Url. xxx.xxx.xxx.xxx	Specifies a logout URL for a specific trusted server. (For example, Security.Logout.Url.<IP address OR domain name>=http://www.novell.com.) Each trusted server can have its own logout URL.
Security.timeout	Specifies the number of minutes of inactivity before the servlet drops a user's session.
Security.Timeout.path	Specifies the directory where timeout information is temporarily stored so that after a successful re-authentication, a users session can be restored.
Security.UseClientIP.enable	Specifies whether the client's IP address to secure sessions: true or false.
Security.UseClientCookie.enable	Specifies whether to allow sessions to be secured using session cookies. This general setting is overridden by individual UI template settings. (For example, Templates.Interface.1.useCookie=true.)
Templates.Accept.[n].id	Specifies the accept string sent by a browser request. (For example, text/x-hdml.) This value is used if the UserAgent string does not match.
Templates.Accept.[n].interface	Specifies which set of templates to use for the identified browser. (For example, simple and wml.)
Templates.Applets.url	Specifies the root of the URL where the applets are located and allows applets to exist on a different server. The default location of applets is beneath the web server's document root directory, so this setting allows you to move the applets to another root directory, but the directory structure (com\novell\webaccess\ . .) beneath the root must remain intact.

Value	Meaning
Templates.Cache.enable	<p>Specifies if templates should be cached:</p> <ul style="list-style-type: none"> ◆ true The template will be compiled the first time it is encountered by the servlet and cached for future access. ◆ false The template is reloaded and compiled each time it is referenced, which is useful if you are modifying templates and want to see changes without restarting the server.
Templates.Cache.size	<p>Specifies the size of the template cache (in kilobytes). See “Adjusting the Cache Size” on page 97. The actual amount of cache space used is big enough to hold the currently cached templates only.</p>
Templates.Debug.enable	<p>Specifies if the templates should be compiled in debug mode (true = Enable debug).</p> <p>When compiled in debug mode, there may be more information available if you're having trouble getting a template to process completely. Some error messages may be more specific about where in the template an error is occurring. Also, when the template is being processed, the template will be dumped to the console window as it processed. This will allow you to see just how far the template processing was able to progress before encountering a problem in the template.</p>
Templates.Images.url	<p>Specifies the root of the URL where the image files are located and allows images to exist on a different server. The default location of images is beneath the web server's document root directory, so this setting allows you to move the images to another root directory, but the directory structure (com\novell\webaccess\ . .) beneath the root must remain intact.</p>
Templates.interface	<p>Specifies the default interface to use (see “Specifying a Template” on page 97). This should be the name of a subdirectory under the directory specified in the Templates.path parameter. For example Templates.interface=frames. This value can be overridden by specifying User.interface when logging in (action=User.Login&User.interface=frames).</p>
Templates.Interface.[n].id	<p>Specifies a specific template directory for the application. (For example, frames and simple.)</p>
Templates.Interface.[n].contentType	<p>Specifies the type of context the templates return to the browser. (For example, text/html or text/xml.)</p>
Templates.Interface.[n].disableCache	<p>Specifies whether to set specific headers in the HTTP response to attempt to disable proxy server and browser cache when using this set of templates: true or false.</p>
Templates.Interface.[n].useCookie	<p>Specifies whether to use session cookies to secure the user's sessions: true or false.</p>
Templates.Java.package	<p>Specifies the Java Package Name used by the template processor to locate files being used as string tables by the templates.</p>
Templates.path	<p>Specifies the path to the template files as a directory name. The directory contains subdirectories that hold the actual template files. Each of these subdirectories corresponds to the user interface implemented by the templates in that directory. For example, if Templates.path has the value \java\servlets\com\novell\webaccess\templates, there could be a directory \java\servlets\com\novell\webaccess\templates\frames which contains the frames user interface.</p>

Value	Meaning
Templates.UserAgent.[n].id	Specifies the string used to identify a browser request. (For example, Windows CE or Blazer.) Used if the interface is not specified on the URL.
Templates.UserAgent.[n].interface	Specifies which set of templates to use for the identified browser. (For example, simple or wml.)
User.Access.calendar	Specifies if users have access to the calendar views (true = allowed).
User.Access.document	Specifies if users have access to document management (true = allowed).
User.Access.HTMLDoc	Specifies whether to allow viewing the documents using the HTML viewers: true or false.
User.Access.HTMLDocNative	Specifies whether to allow opening documents using a native browser plug-in: true or false.
User.Access.internet	Is not currently used.
User.Access.LDAP	Specifies if users have access to LDAP directories (true = allowed).
User.Access.mail	Is not currently used.
User.Access.security	Specifies if users can change their password (true = allowed).
User.Access.speller	Is not currently used.
User.Access.View	Specifies whether to allow viewing attachments using the HTML viewers: true or false.
User.Access.ViewNative	Specifies whether to allow opening attachments using a native browser plug-in: true or false.

6.2 Java Packages

A Java package name is a directory path which uses the period ('.') instead of a slash ('\') to separate components of the path. The package name should specify a path beneath a directory listed in the Java class path. For example, the class path may include `\java\servlets`.

The templates and their string tables can reside in a directory together in the `\java\servlets\com\novell\webaccess\templates\frames` directory, and the resulting package name would be `com.novell.webaccess.templates`.

The templates would refer to the string table named `strings_en.properties` with `{STRINGTABLE frames.strings}`. (The `_en.properties` is appended by the template processor based on the language being used; English in this case. See `STRINGTABLE` in the Template Documentation for more information on string tables.)

6.3 Providers

Providers are Java classes that implement the interface `com.novell.webaccess.common.Provider`.

Once the provider class has been created, it can be registered to the servlet by adding it to the servlet configuration file (see [“Configuration Settings” on page 87](#) for the `Provider.xxx.class`). The provider class will be loaded dynamically by the servlet the first time it is referenced.

When the servlet is installed, two providers are installed with it. For more information about these providers, see

- ♦ [Section 6.3.1, “WebAccess GroupWise Provider,” on page 93](#)
- ♦ [Section 6.3.2, “WebAccess LDAP Provider,” on page 93](#)

6.3.1 WebAccess GroupWise Provider

This provider interfaces with the GroupWise WebAccess Agent (GWINTER) and provides the GroupWise functionality for the servlet. This provider is also used by the servlet for authentication.

6.3.2 WebAccess LDAP Provider

The WebAccess LDAP Provider provides the LDAP functionality for the servlet. It is initially set up to do LDAP searches using BigFoot and SwitchBoard.

6.4 Localization and Internationalization

Localization refers to the language used for the user interface. Internationalization goes beyond localization to ensure that the characters used for a language map correctly to the character set in use by the browser that will be displaying the user interface. The templates installed with the WebAccess Servlet get all language dependent information from string tables. Each language will have string table files included in the directory where the templates exist. The string table files will have names ending with `_xx.properties` where `xx` is the two-letter ISO code for the language. The two letter language code is the same used by `User.lang` that is supplied with the `User.Login` action, or if `User.lang` is not supplied, the `Language.default` setting in the configuration file.

The character mapping is based on information supplied by the browser in the `Accept-Charset` and `Accept-Language` headers.

7 Servlet Tasks

The WebAccess servlet is accessible through HTTP transactions sent from a browser. The requests can be in the form of either the HTTP GET method as well as the POST method. HTML forms can specify a content-type of either application/x-www-form-urlencoded (default content-type when a form is submitted) or multipart/mixed (content-type required to upload files with the <INPUT type="file" . . .> HTML form control).

This section covers the following topics:

- ♦ [Section 7.1, "Configuring the Servlet," on page 95](#)
- ♦ [Section 7.2, "Calling the Servlet," on page 95](#)
- ♦ [Section 7.3, "Specifying a Provider," on page 96](#)
- ♦ [Section 7.4, "Specifying the Action," on page 96](#)
- ♦ [Section 7.5, "Specifying a Template," on page 97](#)

7.1 Configuring the Servlet

- To configure the servlet manager installed with WebAccess, add the following servlet information to the `\java\servlets\servlets.properties` file:

```
servlet.webacc.code=com.novell.webaccess.WebAccessServlet
servlet.webacc.initArgs=Config=sys:/Novell/WebAccess/webacc.cfg
```

The first line names the servlet, and specifies that the Java class that implements the servlet is `com.novell.webaccess.WebAccessServlet`, which corresponds to the file `com\novell\webaccess\WebAccessServlet.class`. This path must exist beneath a directory specified in the Java class path (or servlet class path).

The second line specifies the arguments that are passed to the servlet when it is initialized. The only argument passed to the servlet is `Config=sys:/Novell/WebAccess/webacc.cfg`, which specifies the name of the Servlet's configuration file. All other configuration information that is needed by the servlet is contained within the `\Novell\WebAccess\webacc.cfg` file.

If you are using a servlet manager other than Novell's Servlet Gateway, refer to the documentation for the servlet manager you are using.

7.2 Calling the Servlet

- To call the servlet, specify a URL or HTML form that addresses the HTTP server where the servlet is running and the servlet.

```
<A href="http://111.222.333.4/servlet/webacc"> or
<A href="/servlet/webacc"> if referenced from a page on the server.
```

You can pass additional information by specifying a question mark after the name of the servlet, followed by the data. For example:

```
<A href="/servlet/webacc?User.context=xyz&merge=login">
```

In a form:

```
<FORM method=Post action="/servlet/webacc">
```

You can pass additional information using HTML form controls.

For maximum portability when using different servlet managers, specify only `/servlet/webacc` as the action in a FORM tag. Some servlet managers might not send the data after the question mark. Information that would have gone after the question mark can be added as hidden input fields in the form. For example:

```
<INPUT type="hidden" name="User.context" value="xyz">
<INPUT type="hidden" name="merge" value="login">
```

7.3 Specifying a Provider

Requests that are sent to the servlet either through URLs or HTML forms can instruct the servlet to invoke a specific provider by specifying the name of the provider to invoke.

- ❑ To specify the name of the provider, supply `Provider.name=xxx`, where `xxx` is the name of the provider.

For example, in the `webacc.cfg` config file:

```
Provider.GWAP.class=com.novell.webaccess.providers.XGWAP
```

In a URL:

```
<A href=/servlet/webacc?Provider.name=GWAP&action=Folder.Open& . . .>
```

The provider name is established in the servlet's config file. If `Provider.name` is not specified, the default provider will be invoked.

7.4 Specifying the Action

Actions cause the provider to perform a task and return results.

- ❑ To specify the action the provider should perform, supply `action=xxx`, where `xxx` is the action name.

Also specify any additional parameters that the action needs. For example, all actions except `User.Login` must have the `User.context` parameter specified so that the servlet can associate the request with the session.

Providers should check for the action parameter to determine the action to be performed. However, the servlet cannot ensure that any provider actually uses the action parameter.

7.5 Specifying a Template

After the Servlet and provider have finished performing the action, the results will be merged with a template to produce the final document that is ultimately sent back to the browser or HTTP client.

For more information, see the following topics:

- ♦ [Section 7.5.1, “Specifying the Template If an Error Occurs,” on page 97](#)
- ♦ [Section 7.5.2, “Adjusting the Cache Size,” on page 97](#)

The default template is `login.htt`, which is located in the directory specified by the `Templates.path` setting and the `Templates.interface` (or `User.interface` if it was specified).

- To specify a different template, supply the parameter `merge` along with the name of the template to use (without the `.htt` extension).

For example, `...&merge=file&...`

7.5.1 Specifying the Template If an Error Occurs

- To specify a template to be used in case an error occurs, supply the error parameter along with the name of the template to use (without the `.htt` extension).

For example, `...&merge=file&error=errfile&...`

7.5.2 Adjusting the Cache Size

If you are adding new templates, you may want to adjust the size of the cache.

- To determine the cache size you need, set `Log.level` to `verbose`, set `Templates.Cache.enable` to `true`, and run the servlet.

Each time a template is compiled and added to the cache, a message will indicate how much cache space is in use.

8 Servlet Methods

The following two servlet methods are handled by the servlet and are not passed on to a provider:

- ♦ [“User.Login” on page 100](#)
- ♦ [“User.Logout” on page 101](#)

User.Login

Logs in.

Parameters:

User.id (Required)

Specifies the GroupWise user ID.

User.password (Required)

Specifies the GroupWise password.

GWAP.ip

Specifies the IP address of the GroupWise WebAccess agent (GWINTER) and is used to override the default agent.

GWAP.port

Specifies the port used by the GroupWise WebAccess agent (GWINTER) and is used to override the default agent.

User.context

Specifies the value that was returned from the servlet after making a request that didn't supply a User.context parameter.

User.interface

Specifies the name of the directory (the subdirectory beneath the Templates.path directory provided in the configuration file) that contains the templates that will generate the user interface for this session.

User.lang

Specifies the two-letter ISO code for the desired language for the user interface.

User.Logout

Logs out of WebAccess.

Parameters:

User.context

Specifies the context identifier returned from the User.Login action.

9 Provider Tasks

This section describes the following tasks that are necessary to allow a browser access to documents available to GroupWise:

- ♦ [Section 9.1, “Calling the GroupWise Provider,” on page 103](#)
- ♦ [Section 9.2, “Communicating With the Servlet,” on page 103](#)
- ♦ [Section 9.3, “Specifying a Template,” on page 103](#)

9.1 Calling the GroupWise Provider

- ❑ To call the provider, include in the URL (or in a [hidden] form field), the word “action” accompanied by the name of the action to perform (for example, User.Login). Also include in the URL (or in other form fields) the name of the necessary parameters and their values.

GWAP is the name of the GroupWise provider . When WebAccess is installed, GWAP is specified as the default provider so it is not necessary to specify the name of the provider in the URL (or in forms). If the default provider is changed (in webacc.cfg), you must provide the Provider.name parameter with all requests so that the servlet will call the GroupWise provider.

9.2 Communicating With the Servlet

Actions and their associated parameters are processed by the WebAccess servlet, which is installed when you install GroupWise WebAccess.

- ❑ To call the servlet, create a URL that references the IP address or host name of the computer where the servlet was installed (the computer that is running your HTTP server) and specify /servlet/webacc.

For example,

```
http://151.123.456.78/servlet/webacc
```

9.3 Specifying a Template

When a URL is clicked or a form is submitted, the browser sends the action to an HTTP server, which in turn calls the WebAccess servlet. If action is specified, the servlet calls the provider and then uses a template to generate the resulting HTML page. If no template is specified, login.htt is used.

For more information about how to handle errors, see

- ♦ [Section 9.3.1, “Specifying a Template If an Error Occurs,” on page 104](#)

- ❑ To specify a template, include the word “merge” accompanied by the name of the template to be used.

Templates always end with an .htt extension, but the extension is not included when specifying the name of the template. For example,

```
. . .&merge=webacc
```

uses the webacc.htt template file to generate the HTML that is sent back to the browser.

For more information on templates, see the Template Documentation and the following two examples:

Example 1

```
<A href="/servlet/webacc?Provider.name=GWAP&action=
User.Login&User.id=jdoe&User.password=skiutah&merge=webacc">Login</A>
```

Example 2

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name="Provider.name" value="GWAP">
  <INPUT type=hidden name="merge" value="webacc">
  <INPUT type=hidden name="action" value="User.Login">
  User Id:
  <INPUT type=input name="User.id" value=""><BR>
  Password:
  <INPUT type=password name="User.password" value=""><BR>
</FORM>
```

Since GWAP is the default provider for the WebAccess servlet, Provider.name is not necessary in the previous example.

9.3.1 Specifying a Template If an Error Occurs

- ❑ To specify that a different template be used if an error occurs during processing, include the word "error" accompanied by the name of the template to be used. For example

```
. . .&merge=webacc&error=errfile. . .
```

10 Provider Methods

GroupWise WebAccess Provider interfaces with GroupWise to retrieve information and perform operations.

This document describes the following methods that can be used in URLs and HTML forms to allow access to GroupWise information:

- ♦ [Section 10.1, "Attachment," on page 106](#)
- ♦ [Section 10.2, "Calendar," on page 109](#)
- ♦ [Section 10.3, "Compose," on page 114](#)
- ♦ [Section 10.4, "Directory," on page 156](#)
- ♦ [Section 10.5, "Document," on page 196](#)
- ♦ [Section 10.6, "Find," on page 201](#)
- ♦ [Section 10.7, "Folder," on page 213](#)
- ♦ [Section 10.8, "Library," on page 233](#)
- ♦ [Section 10.9, "Item," on page 237](#)
- ♦ [Section 10.10, "Miscellaneous," on page 277](#)
- ♦ [Section 10.11, "Proxy," on page 279](#)
- ♦ [Section 10.12, "Rules," on page 296](#)
- ♦ [Section 10.13, "Security," on page 311](#)
- ♦ [Section 10.14, "Send Options," on page 313](#)
- ♦ [Section 10.15, "Settings," on page 319](#)
- ♦ [Section 10.16, "Signature," on page 323](#)
- ♦ [Section 10.17, "Time Zone," on page 326](#)
- ♦ [Section 10.18, "User Authentication," on page 330](#)

10.1 Attachment

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Attachment.Save” on page 107](#)
- ◆ [“Attachment.View” on page 108](#)

Attachment.Save

Saves the specified attachment.

Parameters

Item.drn (Required)

Specifies the identifier of the item to which this attachment is associated.

Item.Attachment.id (Required)

Specifies the identifier of the attachment to be saved.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the original contents of the attachment.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Attachment.Save&Item.drn=321z4z&Item.Attachment.id=1"> Save </A>
```

Attachment.View

Views the specified attachment.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Provider.name (Required)

Specifies the name of the GroupWise WebAccess provider: GWAP.

Item.type (Required)

Specifies the type of the item:

AREF HTML representation of the document

HREF Native document (open in local browser plug-in or helper application)

Item.Attachment.id(Required)

Specifies the identifier of the attachment to be saved.

Item.drn (Required)

Specifies the identifier of the item to which this attachment is associated.

Return Values

Returns an HTML version of the attachment, if a conversion is possible.

Remarks

Viewing an attachment requires that some of the parameters be specified in the extra path info attribute of the URL. (The extra path info attribute is the portion of the URL that follows the servlet name and precedes the question mark.)

The parameters must to occur in the following order:

```
/servlet/webacc/User.context/Provider.name/Item.type/Item.Attachment.id
```

If Item.type is AREF, an HTML version of the attachment is returned (if a conversion is possible). If Item.type is HREF, the document is returned in its native format. If an appropriate viewer exists on the user's machine, the document is displayed in that viewer. Otherwise, the user is asked to save the file to disk.

Example

```
<A href="/servlet/webacc/qopkOafrnfAq/GWAP/AREF/1?
action=Attachment.View&Item.Attachment.id=1&User.context=
qopkOafrnfAq&Item.drn=1773z3z1757"> View </A>
```

10.2 Calendar

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Calendar.Search” on page 110](#)

Calendar.Search

Retrieves calendar items (appointments, notes, and tasks) that match the specified criteria.

Parameters

Calendar.queryCount

Specifies how many calendar items to retrieve.

Calendar.startDate

Specifies the start date to use for the calendar query (as the number of milliseconds since 12:00.00 a.m. on January 1, 1970). The default is the current time. This value is not used if Calendar.queryDate is specified or if Calendar.queryType=Calendar.queryDate.

Calendar.dateFormat

(Optional) Specifies the format that the startDate and endDate are in. It must be a valid string pattern as accepted by the Java SimpleDateFormat class.

Calendar.queryType

Specifies the type of calendar search. The default is the current day. Other valid values include the following:

Calendar.nextDay	The next day relative to Calendar.startDate.
Calendar.nextWeek	The next week relative to Calendar.startDate.
Calendar.nextMonth	The next month relative to Calendar.startDate.
Calendar.nextYear	The next year relative to Calendar.startDate.
Calendar.prevDay	The previous day relative to Calendar.startDate.
Calendar.prevWeek	The previous week relative to Calendar.startDate.
Calendar.prevMonth	The previous month relative to Calendar.startDate.
Calendar.nextYear	The next year relative to Calendar.startDate.
Calendar.queryDate	A specific date is desired. If specified, the following parameters must also be specified: Calendar.queryDay, Calendar.queryMonth, and Calendar.queryYear.

Calendar.queryDate

Specifies that a specific date is desired. This value is ignored if Calendar.queryType is specified. The value of this parameter is ignored. If Calendar.queryDate is specified, the following parameters must be present:

Calendar.queryDay
Calendar.queryMonth
Calendar.queryYear

Calendar.queryDay

Specifies the desired day of the month.

Calendar.queryMonth

Specifies the desired month of the year (1 = January, 12 = December).

Calendar.queryYear

Specifies the desired year (in four-digit format, such as 1999 or 2000).

Calendar.nextDay

Specifies the day after the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.nextWeek

Specifies the day one week after the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.nextMonth

Specifies the month one month after the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.nextYear

Specifies the day one year after the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.prevDay

Specifies the day before the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.prevWeek

Specifies the day one week before the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.prevMonth

Specifies the day one month before the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.prevYear

Specifies the day one year before the day specified by Calendar.startDate. The current date is used if Calendar.startDate is not specified.

Calendar.endDate

Specifies the time (or the number of milliseconds since 12:00.00 a.m. January 1, 1970) after which to stop retrieving items.

Calendar.durationType

Specifies the time duration for which items should be retrieved (ignored if Calendar.endDate is specified):

Day Returns items for one day (Default)

Week Returns items for one week

Month Returns items for one month

Calendar.queryMessageType

Specifies the type of messages to retrieve (defaults to receive all appointments, notes, and tasks):

Appointment

Note

Task

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Calendar.firstDayOfWeek	Day of the week that the month of the starting date falls on (1 = Sunday, 2 = Monday, . . . 7 = Saturday).
Calendar.dayCount	Number of days in the month of the starting date.
Calendar.endDate	Ending date (represents the number of milliseconds since 12:00.00 a.m. January 1, 1970).
Calendar.startDate	Starting date (represents the number of milliseconds since 12:00.00 a.m. January 1, 1970).
Folder.hasMore	1 = There are more messages than what was returned.
Item.count	Number of items returned.
Item[n].id	<i>n</i> th item within the list of items returned.
Item[n].drn	GroupWise identifier of the <i>n</i> th item returned.
Item[n].type	Type of the <i>n</i> th item: Appointment, Note, or Task.
Item[n].priority	Priority of the <i>n</i> th item.
Item[n].from	Sender of the <i>n</i> th item.
Item[n].told	List of user identifiers to whom the <i>n</i> th item was sent.
Item[n].subject	Subject of the <i>n</i> th item.
Item[n].subType	Subtype of the <i>n</i> th item: Item.SHARED_ADDRESS_BOOK Item.PERSONAL Item.SHARED_FOLDER Item.DISCUSSION Item.WORKFLOW Item.REPLY_REQUESTED
Item[n].Status.read	Read or unread.
Item[n].Status.accepted	Accepted or unaccepted.
Item[n].Status.completed	Completed or uncompleted.
Item[n].dateSent	Date the <i>n</i> th item was sent.
Item[n].dateStart	Start time for the <i>n</i> th item (for calendar items).
Item[n].dateEnd	End time for the <i>n</i> th item (for appointments, due date for tasks).
Item[n].location	Location of the <i>n</i> th item (for appointments).

Value	Description
Item[n].isOutbound	<i>n</i> th item is a sent item.
Item[n].Priority.task	Priority of the <i>n</i> th item.
Item[n].dateTaskAssigned	Assigned date for the <i>n</i> th item (for tasks).
Item[n].hasAttachment	1 = <i>n</i> th item has attachments.
Folder.hasInbound	1 = returned list of items contains received items.
Folder.hasOutbound	1 = returned list of items contains send items.

Example

```

<! Search for today's calendar items >
<A href="/servlet/webacc?User.context=xyz&action=Calendar.Search">Today</A>

<! Search for tomorrow's calendar items >
<A href="/servlet
webacc?User.context=xyz&action=Calendar.Search&queryType=Calendar.queryNext">Tomor
row</A>

<! Search for Calendar Items for 3 January 2000 >
<A href="/servlet
webacc?User.context=xyz&action=Calendar.Search&Calendar.queryType=queryDate&Calend
ar.queryDa
=3&Calendar.queryMonth=1&Calendar.queryYear=2000">January 3, 2000</A>
<! OR >
<A href="/servlet/webacc?User.context=xyz&action=Calendar.Search&Calendar.
QueryDate=&Calendar.queryDay=3&Calendar.queryMonth=1&Calendar.queryYear=2000">Janu
ary 3, 2000</A>

```

10.3 Compose

Click any action name to view the description, parameters, and return values for that action.

- ♦ [“Compose.Action” on page 115](#)
- ♦ [“Compose.Attach” on page 123](#)
- ♦ [“Compose.Attach.Delete” on page 126](#)
- ♦ [“Compose.BC” on page 129](#)
- ♦ [“Compose.BusySearch” on page 130](#)
- ♦ [“Compose.Cancel” on page 133](#)
- ♦ [“Compose.CC” on page 134](#)
- ♦ [“Compose.ChangeTo.xxx” on page 135](#)
- ♦ [“Compose.Delegate” on page 136](#)
- ♦ [“Compose.Directory” on page 139](#)
- ♦ [“Compose.FolderSel” on page 140](#)
- ♦ [“Compose.Forward” on page 141](#)
- ♦ [“Compose.Reply” on page 144](#)
- ♦ [“Compose.SaveDraft” on page 147](#)
- ♦ [“Compose.Send” on page 148](#)
- ♦ [“Compose.SendOptions.Modify” on page 149](#)
- ♦ [“Compose.SendOptions.Read” on page 151](#)
- ♦ [“Compose.To” on page 153](#)

Compose.Action

Performs a compose action specified by the sub method.

Parameters

For parameter information, see the documentation for each of the following subcommands:

[Compose.Attach \(page 123\)](#)
[Compose.Attach.Delete \(page 126\)](#)
[Compose.BC \(page 129\)](#)
[Compose.BusySearch \(page 130\)](#)
[Compose.Cancel \(page 133\)](#)
[Compose.CC \(page 134\)](#)
[Compose.ChangeTo.xxx \(page 135\)](#)
[Compose.Delegate \(page 136\)](#)
[Compose.Directory \(page 139\)](#)
[Compose.Forward \(page 141\)](#)
[Compose.Reply \(page 144\)](#)
[Compose.Send \(page 148\)](#)
[Compose.SendOptions.Modify \(page 149\)](#)
[Compose.SendOptions.Read \(page 151\)](#)
[Compose.To \(page 153\)](#)

To begin composing a new item, Specify Compose.Action without any subcommands.

Compose.id

Specifies the ID of this composition.

Calendar.queryYear

Specifies the year for the appointments.

Calendar.queryMonth

Specifies the month for the appointments.

Calendar.queryDay

Specifies the day of the month for the appointment.

Calendar.queryAM

Specifies 1 for a.m. appointments and 0 for p.m.

Calendar.queryDuration

Specifies the duration of appointment: number of minutes, hours, or days as specified by Calendar.queryDurationType.

Calendar.queryDurationType

Specifies the duration type for appointments: 1 =minutes, 2 = hours, 3 = days.

Calendar.queryTime

Specifies the appointment time, such as 2:00 p.m.

Item.Compose.method

Specifies if the compose item is a posted item:

POST indicates a posted item

Item.bc

Specifies the list of BC recipients.

Item.caller

Specifies the caller if the item type is Phone.

Item.cc

Specifies the list of CC recipients.

Item.company

Specifies the company if the item type is Phone.

Item.dateEnd

Specifies the end date if an item is an appointment or the due date if the item is a task.

Item.dateStart

Specifies the start date if an item is an appointment or a note.

Item.duration

Specifies the duration if the item is an appointment.

Item.location

Specifies the location of the appointment.

Item.message

Specifies the message text.

Item.phone

Specifies the phone number if the item type is Phone.

Item.priority

Specifies the Item priority: Low, Medium, High.

Item.ReplyRequested.day

Specifies the number of days selected for a reply request.

Item.ReplyRequest.type

Specifies the type of reply requested: WhenConvenient or ByDate.

Item.ReturnNotification.accepted

Specifies the type of return notification for an accepted item:

1 None

2 Mail

4 Notify

8 Notify and Mail

Item.ReturnNotification.completed

Specifies the type of return notification for a completed item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.declined

Specifies the type of return notification for a declined item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.deleted

Specifies the type of return notification for a deleted item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.opened

Specifies the type of return notification for an opened item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.security

Specifies the security settings on the item.

Item.taskPriority

Specifies the priority of an item if the item is a task.

Item.Task.startDate

Specifies the start date if the item is a task.

Item.to

Specifies the list of To recipients.

Item.type

Specifies the item type: Mail, Appointment, Task, Note, or Phone.

Item.Accept.ShowAs

Specifies one of the following acceptance levels for an appointment:

- 1 Busy
- 2 Free
- 4 Out of office

NOTE: This parameter is used with GroupWise 7 SP 1 and later versions only.

Return Values

Some or all of the following values might be returned. None of them are required, however, or guaranteed to be always be returned.

Value	Description
Compose.id	Specifies specific item being composed.
Item.areConflicts	Specifies if there are scheduling conflicts (appointments).
Item.bc	Specifies the list of BC recipients.
Item.bcName	Specifies the list of BC recipient display names.
Item.caller	Specifies the caller if the item type is Phone.
Item.cc	Specifies the list of CC recipients.
Item.ccFullId	Specifies the list of CC recipient full IDs.
Item.ccName	Specifies the list of CC recipient display names.
Item.Child.depth	Specifies the depth level of the child item.
Item.Child.id	Specifies the ID of the child item.
Item.company	Specifies the company if the item type is Phone.
Item.Compose.method	Specifies if the compose item is a posted item: POST indicates a posted item.
Item.dateEnd	Specifies the end date if the item is an appointment or the due date if the item is a task.
Item.dateSent	Specifies the date on which the item was sent.
Item.dateStart	Specifies the start date if the item is an appointment or note.
Item.Document.ext	Specifies the document.
Item.Document.id	Specifies the document ID (the unique identifier of the document).
Item.Document.version	Specifies the document version.
Item.drn	Specifies the unique identifier of the item.
Item.duration	Specifies the duration if the item is an appointment.
Item.durationType	Specifies the duration type for an appointment: 1 = Minutes, 2 = Hours, 3 = Days.
Item.Folder.newName	Specifies the new name of the folder.
Item.from	Specifies the From text of the item.
Item.fromFullId	Specifies the Full FromID, which is the user ID of the from text.
Item.id	Specifies the index of the item within the current list of items.
Item.isAutoDate	Specifies if the item was an autodate (appointments/tasks).

Value	Description
Item.isDraft	Specifies the item is a draft item.
Item.isOutbound	Specifies if the item is a Sent Item.
Item.Library.id	Specifies the document library ID.
Item.location	Specifies the location of the appointment.
Item.message	Specifies the Message Text of the item.
Item.modified	Specifies if the compose item has unsaved changes.
Item.phone	Specifies the phone number if the item type is Phone.
Item.priority	Specifies the priority of the item: Low, Medium, High.
Item.ReplyRequested.date	Specifies the date by which the recipient should respond.
Item.ReplyRequested.day	Specifies the number of days that were selected for a reply request.
Item.ReplyRequested.type	Specifies the type of reply that is requested: When Convenient or By Date.
Item.ReturnNotification. accepted	Specifies the type of return notification for an accepted item: 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.ReturnNotification. completed	Specifies the type of return notification for a completed item. 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.ReturnNotification. declined	Specifies the type of return notification for a declined item. 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.ReturnNotification. deleted	Specifies the type of return notification for a deleted item. 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.ReturnNotification. opened	Specifies the type of return notification for an opened item. 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.security	Specifies the security settings on the item.
Item.status	Specifies the status of the item.

Value	Description
Item.Status.accepted	Specifies if the item has been accepted (appointments and tasks).
Item.Status.completed	Specifies if the item has been completed (tasks).
Item.Status.isDeleted	Specifies if the item has been deleted.
Item.Status.read	Specifies the read status of the message.
Item.subject	Specifies the Subject text.
Item.subtype	Specifies the subtype of the item.
Item.taskPriority	Specifies the priority of the item if the item is a Task.
Item.Task.startDate	Specifies the start date if the item is a task.
Item.to	Specifies the list of To recipients.
Item.toFullId	Specifies the list of To recipient full IDs.
Item.toName	Specifies the list of To recipient display names.
Item.told	Specifies the list of To recipient user IDs.
Item.type	Specifies the type of the item: Appointment, Note, Phone, or Task.
Signature.isAutomatic	Specifies if the signature should be automatically added to the message.
Signature.isEnabled	Specifies a signature exists and is enabled.
Signature.signature	Specifies the GroupWise Signature Text.

If the item is being posted (such as a personal item or discussion item), the following might also be returned:

Value	Description
Item.Folder.id	Specifies the ID of the folder the item is posted to.
Item.Folder.name	Specifies the name of the folder the item is posted to.
Item.Folder.type	Specifies the type of the folder that the item is posted to.

If the item has an enclosure (such as the original message when forwarding or delegating a message), the following might also be returned:

Value	Description
Item.Enclosure.id	Specifies the identifier of the enclosed item.
Item.Enclosure.drn	Specifies the GroupWise specific identifier of the enclosed item.
Item.Enclosure.type	Specifies the item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Specifies the sender of the enclosed item.
Item.Enclosure.fromFullId	Specifies the full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Specifies the priority of the enclosed item.

Value	Description
Item.Enclosure.to	Specifies the list of To recipients for the enclosed item.
Item.Enclosure.cc	Specifies the list of CC recipients for the enclosed item.
Item.Enclosure.bc	Specifies the list of BC recipients for the enclosed item.
Item.Enclosure.told	Specifies the user IDs of the To recipients.
Item.Enclosure.toFullId	Specifies the full resolved IDs of the To recipients.
Item.Enclosure.toName	Specifies the names of the To recipients.
Item.Enclosure.ccid	Specifies the user IDs of the CC recipients.
Item.Enclosure.ccFullId	Specifies the full resolved IDs of the CC recipients.
Item.Enclosure.ccName	Specifies the names of the To recipients.
Item.Enclosure.subject	Specifies the subject of the enclosed item.
Item.Enclosure.message	Specifies the message of the enclosed item.
Item.Enclosure.subType	Specifies the subtype of the enclosed item.
Item.Enclosure.Folder.newName	Specifies the new name of the folder.
Item.Enclosure.read	Specifies the read status of the enclosed item.
Item.Enclosure.accepted	Specifies the accepted status of the enclosed item.
Item.Enclosure.completed	Specifies the completed status of the enclosed item.
Item.Enclosure.Child.id	Specifies the ID of the enclosed item.
Item.Enclosure.Child.depth	Specifies the depth of the enclosed item and indicates how deeply nested the enclosure is. For example, if you forward a forwarded item, the original item has a depth of 2.
Item.Enclosure.dateSent	Specifies the date the enclosed item was sent.
Item.Enclosure.dateStart	Specifies the start date of the enclosed item.
Item.Enclosure.dateEnd	Specifies the date or due date of the enclosed item.
Item.Enclosure.ReplyRequest.type	Specifies the type of reply requested for the enclosed item.
Item.Enclosure.ReplyRequest.date	Specifies the date associated with the reply request of the enclosed item.
Item.Enclosure.location	Specifies the location of the enclosed appointment item.
Item.Enclosure.Task.priority	Specifies the priority of the enclosed item.
Item.Enclosure.Task.startDate	Specifies the start date of the enclosed task item.
Item.Enclosure.caller	Specifies the caller of the enclosed phone item.
Item.Enclosure.company	Specifies the company of the enclosed phone item.
Item.Enclosure.phone	Specifies the phone of the enclosed phone item.
Item.Enclosure.isAutoDate	Specifies 1 if the enclosed item is an auto-date item.
Item.Enclosure.security	Specifies the security of the enclosed item.

Value	Description
Item.Enclosure.status	Specifies the status of the enclosed item: read, deleted, accepted, etc.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Compose.Action&merge=send">Create New Message</A>
```

Compose.Attach

Attaches a file to this compose item. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the item being composed.

Item.Attachment.name (Required)

Specifies the name of the file to be attached.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Item.Attachment.count	Number of attachments this item contains.
Item.Attachment[n].id	Identifier of the <i>n</i> th attachment.
Item.Attachment[n].name	Name of the <i>n</i> th attachment.
Item.Attachment[n].type	Type of the <i>n</i> th attachment.
Item.Attachment[n].size	Size of the <i>n</i> th attachment.
Item.Attachment[n].filename	File name of the <i>n</i> th attachment.
Item.Attachment[n].category	Category of the <i>n</i> th attachment.
Item.Attachment[n].mimeType	Mime type of the <i>n</i> th attachment.
Item.Attachment[n].Library.id	Library identifier of the <i>n</i> th attachment (document reference).
Item.Attachment[n].Document.id	Document identifier of the <i>n</i> th attachment (document reference).
Item.Attachment[n].Document.version	Document version of the <i>n</i> th attachment (document reference).
Compose.id	ID of this composition.
Item.type	Type of the item: Mail, Appointment, Task, Note, or Phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment or the due date if the item is a task.
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item type is a task.

Value	Description
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item type is Phone.
Item.company	Company if the item type is Phone.
Item.phone	Phone number if the item type is Phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User IDs of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.cclid	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.

Value	Description
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.
Item.Enclosure.isAutoDate	1 if the enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Remarks

When the servlet uploads a file, it creates additional name value pairs from the File input field. The new pairs are created by adding .name and .size to the end of the value of the Name attribute of the File input field. The .name variable is assigned the name of the uploaded file, and the .size is the size of the uploaded file.

In the example, if the user enters a file name of myfile.dat in the file input field, the servlet will create the following two name value pairs in addition to the others specified in the form:

```
Item.Attachment.name=myfile.dat
Item.Attachment.size=58734
```

NOTE: To make the browser upload the file, the EncType attribute of the form tag must have a value of multipart/form-data, as shown in the following:

```
<FORM EncType="multipart/form-data">
```

Example

```
<FORM METHOD=POST ACTION="/servlet/webacc" EncType="multipart/form-data">
  Attach File: <INPUT Type="file" Name="Item.Attachment">
  <INPUT Type=submit Name="Compose.Attach.Add" Value="Add">
  <INPUT type="hidden" name="User.context" value="eInqZdgimsEm">
  <INPUT type="hidden" name="action" value="Compose.Action">
  <INPUT type="hidden" name="Compose.Attach" value="">
  <INPUT type="hidden" name="Compose.id" value="1">
  <INPUT type="hidden" name="merge" value="send">
  <INPUT type="hidden" name="Url.Enclosure.type" value="">
</FORM>
```

Compose.Attach.Delete

Removes an attached file from this compose item. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the item being composed.

Item.Attachment.Id (Required)

Specifies the attachment to be removed from the item being composed.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Item.Attachment.count	Number of attachments this item contains.
Item.Attachment[n].id	Identifier of the <i>n</i> th attachment.
Item.Attachment[n].name	Name of the <i>n</i> th attachment.
Item.Attachment[n].type	Type of the <i>n</i> th attachment.
Item.Attachment[n].size	Size of the <i>n</i> th attachment.
Item.Attachment[n].filename	File name of the <i>n</i> th attachment.
Item.Attachment[n].category	Category of the <i>n</i> th attachment.
Item.Attachment[n].mimeType	Mime type of the <i>n</i> th attachment.
Item.Attachment[n].Library.id	Library identifier of the <i>n</i> th attachment (document reference).
Item.Attachment[n].Document.id	Document identifier of the <i>n</i> th attachment (document reference).
Item.Attachment[n].Document.version	Document version of the <i>n</i> th attachment (document reference).
Compose.id	ID of this composition.
Item.type	Type of the item: Mail, Appointment, Task, Note, or Phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment (or due date if the item is a task).
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item type is a task.

Value	Description
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item is a phone.
Item.company	Company if the item is a phone.
Item.phone	Phone number if the item type is a phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User IDs of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.ccid	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.

Value	Description
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.
Item.Enclosure.isAutoDate	1 if the enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Example

```
<FORM METHOD=POST ACTION="/servlet/webacc">
  <INPUT Type=submit Name="Compose.Attach.Delete" Value="Remove">
  <INPUT Type="Checkbox" Name="Item.Attachment.id" Value="1"> myfile.dat
<BR>
  <INPUT type="hidden" name="User.context" value="elnqZdgimsEm">
  <INPUT type="hidden" name="action" value="Compose.Action">
  <INPUT type="hidden" name="Compose.Attach" value="">
  <INPUT type="hidden" name="Compose.id" value="1">
  <INPUT type="hidden" name="merge" value="send">
</FORM>
```


Compose.BC

Adds recipients to the specified composition as blind copy (BC) recipients, which allows recipients to be added from other sources besides the keyboard. For example, from a directory or address book. This action must be specified as a subcommand of [Compose.Action \(page 115\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the item being composed.

Directory.Item.email (Required)

Specifies where to send this item. This parameter can be repeated so that multiple recipients can be specified with one request.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Compose.Action \(page 115\)](#).

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT id=cb1 type="checkbox" name=Directory.Item.email
    value="jdoe">
  <INPUT id=cb2 type="checkbox" name=Directory.Item.email
    value="jsmith">
  <INPUT Type=submit name=Compose.To value="To">
  <INPUT Type=submit name=Compose.CC Value="CC">
  <INPUT Type=submit name=Compose.BC value="BC">
</FORM>
```

Compose.BusySearch

Performs a busy search for the specified appointment (valid for appointment item types only). This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the identifier of the appointment item being composed, as returned from [Compose.Action](#).

BusySearch.wait

Specifies how many minutes to allow for completion of the busy search. Defaults to the value for `Provider.GWAP.BusySearch.timeOut` from the `WebAccess` config file or 1 minute (if that setting is not in the configuration file).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Item.BusySearch.FreeTimes.count	Number of free blocks of time that were found.
Item.BusySearch.FreeTimes[n].start	Starting time for the <i>n</i> th block of free time.
Item.BusySearch.FreeTimes[n].end	Ending time for the <i>n</i> th block of free time.
Item.BusySearch.Attendee.NotFound.count	Number of attendees for which no busy search occurred.
Item.BusySearch.Attendee.NotFound[n]	Name of the <i>n</i> th attendee that was not found.
Item.BusySearch.Attendee.Found.count	Number of attendees for which the busy search occurred.
Item.BusySearch.Attendee.Found[n]	Name of the <i>n</i> th found attendee.
Compose.id	ID of this composition.
Item.type	Type of the item: Mail, Appointment, Task, Note, or Phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment or the due date if the item is a task.
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item is a task.
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item type is Phone.

Value	Description
Item.company	Company if the item type is Phone.
Item.phone	Phone number if the item type is Phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User IDs of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.cclId	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.

Value	Description
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.
Item.Enclosure.isAutoDate	1 if the enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Example

```

<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  . . .
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  . . .
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT type=submit name=Compose.Directory value="Address Book">
  <INPUT type=submit name=Compose.BusySearch value="Busy Search">
  <INPUT type=submit name=Compose.Cancel value="Cancel">
</FORM>

```

Compose.Cancel

Cancels the compose and releases the memory used for composing the message. This action must be specified as a subcommand of [Compose.Action \(page 115\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the appointment item being composed, as returned from [Compose.Action](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

None.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  . . .
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  . . .
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT type=submit name=Compose.Directory value="Address Book">
  <INPUT type=submit name=Compose.BusySearch value="Busy Search">
  <INPUT type=submit name=Compose.Cancel value="Cancel">
</FORM>
```

Compose.CC

Adds recipients to the specified composition as carbon copy recipients, which allows recipients to be added from other sources besides the keyboard (for example, from a directory or address book). This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the identifier of the item being composed.

Directory.Item.email (Required)

Specifies where to send this item. This parameter can be repeated so that multiple recipients can be specified with one request.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as ["Compose.Action"](#) on page 115.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT id=cb1 type="checkbox" name=Directory.Item.email
    value="jdoe">
  <INPUT id=cb2 type="checkbox" name=Directory.Item.email
    value="jsmith">
  <INPUT Type=submit name=Compose.To value="To">
  <INPUT Type=submit name=Compose.CC Value="CC">
  <INPUT Type=submit name=Compose.BC value="BC">
</FORM>
```

Compose.ChangeTo.xxx

Changes this item to an item of type xxx. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Subcommands

Compose.ChangeTo.Appointment
Compose.ChangeTo.Mail
Compose.ChangeTo.Note
Compose.ChangeTo.Phone
Compose.ChangeTo.Task

Parameters

Compose.id (Required)

Specifies the identifier of the item being composed.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Compose.Action](#) (page 115).

Example

```
<FORM method=post action=/servlet/webacc">
  <INPUT type="hidden" name=User.context value=xyz>
  <INPUT type="hidden" name="action" value=Compose.Action>
  <INPUT type="hidden" name="Compose.id" value=1>
  .
  .
  .
  <INPUT TYPE="Submit" Name="Compose.ChangeTo.Mail"
    Value="Appointment">
  .
  .
  .
</FORM>
```

Compose.Delegate

Sends a delegated item. This action must be specified as a subcommand of the [Compose.Action](#) (page 115) method.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Compose.id	ID of this composition.
Item.action.folderUpdate	If a draft item was sent, indicates that the folder list view might need to be updated.
Item.type	Type of the item: Mail, Appointment, Task, Note, or Phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment or the due date if the item is a task.
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item is a task.
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item type is Phone.
Item.company	Company if the item type is Phone.
Item.phone	Phone number if the item type is Phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.

Value	Description
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User IDs of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.cclid	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.
Item.Enclosure.isAutoDate	The enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT Type=image Name=Compose.Delegate alt="{STRING
    Compose.Image.AltText.send}" src="{VAR Templates.Images.url}
    /btnsend{VAR User.lang}.gif" border=0><BR>
</FORM>
```

Compose.Directory

Lists the GroupWise Directories (address books). This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

In addition to the parameters used by [“Directory.List”](#) on page 182, it might be helpful to pass the `Compose.id` into the process (template) that is creating the directory so that it is available for the directory to add recipients.

Compose.id

Specifies the identifier of the appointment item being composed as returned from `Compose.Action`.

Return Values

Returns the same values as [“Compose.Action”](#) on page 115.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  . . .
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  . . .
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT type=submit name=Compose.Directory value="Address Book">
  <INPUT type=submit name=Compose.BusySearch value="Busy Search">
  <INPUT type=submit name=Compose.Cancel value="Cancel">
</FORM>
```

Compose.FolderSel

Retrieves a list of folders. Can also be used to select a folder in which to save a draft item.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Folder.List \(page 223\)](#).

Compose.Forward

Begins composing a forwarded item. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Compose.id	ID of this composition.
Item.action.folderUpdate	If a draft item was sent, indicates that the folder list view might need to be updated.
Item.type	Type of the item: mail, appointment, task, note, or phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment or the due date if the item is a task.
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item is a task.
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item type is Phone.
Item.company	Company if the item type is Phone.
Item.phone	Phone number if the item type is Phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.

Value	Description
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User ID's of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.cclid	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.
Item.Enclosure.isAutoDate	The enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT Type=image Name=Compose.Forward alt="{STRING
    Compose.Image.AltText.send}" src="{VAR Templates.Images.url}/
    btnsend{VAR User.lang}.gif" border=0><BR>
</FORM>
```

Compose.Reply

Begins composing a reply to an item. This action must be specified as a subcommand of the [Compose.Action \(page 115\)](#) method.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Compose.id	ID of this composition.
Item.action.folderUpdate	If a draft item was sent, indicates that the folder list view might need to be updated.
Item.type	Type of the item: Mail, Appointment, Task, Note, or Phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment or the due date if the item is a task.
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item is a task.
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item type is Phone.
Item.company	Company if the item type is Phone.
Item.phone	Phone number if the item type is Phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.

Value	Description
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User IDs of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.cclid	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.
Item.Enclosure.isAutoDate	Enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT Type=image Name=Compose.Reply alt="{STRING
    Compose.Image.AltText.send}" src="{VAR Templates.Images.url}/
    btnsend{VAR User.lang}.gif" border=0><BR>
</FORM>
```

Compose.SaveDraft

Saves the item as a draft item. This action must be specified as a subcommand of [Compose.Action \(page 115\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the appointment item being composed, as returned from [Compose.Action](#) or [Item.Compose \(page 249\)](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Compose.Action \(page 115\)](#).

Compose.Send

Sends the item. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the identifier of the appointment item being composed, as returned from [Compose.Action](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as ["Compose.Action"](#) on page 115.

If the sent item is a draft item, this method also returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  .
  .
  <INPUT type=input name=Item.to value="jdoe, jsmith">
  .
  .
  <INPUT type=submit name=Compose.Send value="Send">
  <INPUT type=submit name=Compose.Directory value="Address Book">
  <INPUT type=submit name=Compose.BusySearch value="Busy Search">
  <INPUT type=submit name=Compose.Cancel value="Cancel">
</FORM>
```

Compose.SendOptions.Modify

Modifies the current item send options. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the identifier of the appointment item being composed as returned from `Compose.Action`.

Item.priority

Specifies the priority of the item: Low, Medium, or High.

Item.security

Specifies the security settings on the item:

- 1 Normal
- 2 Proprietary
- 3 Confidential
- 4 Secret
- 5 Top Secret
- 6 For your eyes only

Item.ReplyRequested.day

Specifies the number of days before a reply request is expected.

Item.ReplyRequest.type

Specifies the type of reply requested: `WhenConvenient` or `ByDate`.

Item.ReturnNotification.accepted

Specifies the type of return notification for an accepted item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.completed

Specifies the type of return notification for a completed item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.deleted

Specifies the type of return notification for a deleted item:

- 1 None
- 2 Mail

- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.declined

Specifies the type of return notification for a declined item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.opened

Specifies the type of return notification for an opened item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Return Values

Returns the same values as [“Compose.Action” on page 115](#).

Example

```
<FORM method=post action="/servlet/webacc">  
  <INPUT type=hidden name=User.context value=xyz>  
  <INPUT type=hidden name=action value=Compose.Action>  
  <INPUT type=hidden name=Compose.id value=1>  
  . . .  
  <INPUT type=submit name="Item.security" value="High">  
  
  . . .  
  <INPUT type=submit name=Compose.SendOptions.Modify value="1">  
</FORM>
```

Compose.SendOptions.Read

Reads the current item send options. This action must be specified as a subcommand of [Compose.Action](#) (page 115).

Parameters

Compose.id (Required)

Specifies the identifier of the appointment item being composed as returned from `Compose.Action`.

Item.priority

Specifies the priority of the item: Low, Medium, or High.

Item.security

Specifies the security settings on the item:

- 1 Normal
- 2 Proprietary
- 3 Confidential
- 4 Secret
- 5 Top Secret
- 6 For your eyes only

Item.ReplyRequested.day

Specifies the number of days before a reply request is expected.

Item.ReplyRequest.type

Specifies the type of reply requested: `WhenConvenient` or `ByDate`.

Item.ReturnNotification.accepted

Specifies the type of return notification for an accepted item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.completed

Specifies the type of return notification for a completed item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.deleted

Specifies the type of return notification for a deleted item:

- 1 None
- 2 Mail

- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.declined

Specifies the type of return notification for a declined item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

Item.ReturnNotification.opened

Specifies the type of return notification for an opened item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [“Compose.Action” on page 115](#).

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  . . .
  <INPUT type=submit name="Item.security" value="High">
  . . .
  <INPUT type=submit name=Compose.SendOptions.Read value="1">
</FORM>
```


Compose.To

Adds recipients a subcommand of the [Compose.Action \(page 115\)](#) method) to the specified composition as To recipients, which allows recipients to be added from other sources besides the keyboard (from a directory or address book, for example). This action must be specified as a subcommand of [Compose.Action \(page 115\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the item being composed.

Directory.Item.email

Specifies the recipient to send this item to. This parameter can be repeated so that multiple recipients can be specified with one request.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Compose.id	ID of this composition.
Item.type	Type of the item: mail, appointment, task, note, or phone.
Item.to	List of To recipients.
Item.cc	List of CC recipients.
Item.bc	List of BC recipients.
Item.dateStart	Start date if the item is an appointment or note.
Item.dateEnd	End date if the item is an appointment or the due date if the item is a task.
Item.location	Location of the appointment.
Item.taskPriority	Priority of an item if the item is a task.
Item.Task.startDate	Start date if the item is a task.
Item.caller	Caller if the item type is Phone.
Item.company	Company if the item type is Phone.
Item.phone	Phone number if the item type is Phone.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for an appointment: 1=minutes, 2=hours, 3=days.

If the item has an enclosure, such as the original message when forwarding or delegating a message, the following might also be returned:

Value	Description
Item.Enclosure.id	Enclosed item.
Item.Enclosure.drn	GroupWise identifier of the enclosed item.
Item.Enclosure.type	Item type of the enclosure: Mail, Appointment, Note, Task, or Phone.
Item.Enclosure.from	Sender of the enclosed item.
Item.Enclosure.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.Enclosure.priority	Priority of the enclosed item.
Item.Enclosure.told	User IDs of the To recipients of the enclosed item.
Item.Enclosure.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.Enclosure.toName	Display names of the To recipients of the enclosed item.
Item.Enclosure.cclid	User IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.Enclosure.ccName	Display names of the To recipients of the enclosed item.
Item.Enclosure.subject	Subject of the enclosed item.
Item.Enclosure.message	Message of the enclosed item.
Item.Enclosure.subType	Subtype of the enclosed item.
Item.Enclosure.read	Read status of the enclosed item.
Item.Enclosure.accepted	Accepted status of the enclosed item.
Item.Enclosure.completed	Completed status of the enclosed item.
Item.Enclosure.Child.id	Enclosed ID of the enclosed item.
Item.Enclosure.Child.depth	How deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Enclosure.dateSent	Send date of the enclosed item.
Item.Enclosure.dateStart	Start date of the enclosed item.
Item.Enclosure.dateEnd	End or due date of the enclosed item.
Item.Enclosure.ReplyRequest.type	Type of the reply request of the enclosed item.
Item.Enclosure.ReplyRequest.date	Date associated with the reply request of the enclosed item.
Item.Enclosure.location	Location of the enclosed appointment item.
Item.Enclosure.Task.priority	Priority of the enclosed item.
Item.Enclosure.Task.startDate	Start date of the enclosed task item.
Item.Enclosure.caller	Caller of the enclosed phone item.
Item.Enclosure.company	Company of the enclosed phone item.
Item.Enclosure.phone	Phone number of the enclosed phone item.

Value	Description
Item.Enclosure.isAutoDate	Enclosed item is an auto-date item.
Item.Enclosure.security	Security of the enclosed item.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Compose.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT id=cb1 type="checkbox" name=Directory.Item.email
    value="jdoe">
  <INPUT id=cb2 type="checkbox" name=Directory.Item.email
    value="jsmith">
  <INPUT Type=submit name=Compose.To value="To">
  <INPUT Type=submit name=Compose.CC Value="CC">
  <INPUT Type=submit name=Compose.BC value="BC">
</FORM>
```

10.4 Directory

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Directory.Accumulate.Group” on page 157](#)
- ◆ [“Directory.Accumulator.Add.BC” on page 158](#)
- ◆ [“Directory.Accumulator.Add.CC” on page 159](#)
- ◆ [“Directory.Accumulator.Add.To” on page 160](#)
- ◆ [“Directory.Accumulator.Delete” on page 161](#)
- ◆ [“Directory.Accumulator.List” on page 162](#)
- ◆ [“Directory.Action” on page 163](#)
- ◆ [“Directory.CollapseGroup” on page 165](#)
- ◆ [“Directory.CopyEntries” on page 166](#)
- ◆ [“Directory.Create” on page 167](#)
- ◆ [“Directory.CreateEntry” on page 168](#)
- ◆ [“Directory.CreateGroup” on page 171](#)
- ◆ [“Directory.Delete” on page 174](#)
- ◆ [“Directory.DeleteEntry” on page 175](#)
- ◆ [“Directory.GetLastSearch” on page 176](#)
- ◆ [“Directory.HoldEntries” on page 178](#)
- ◆ [“Directory.Info” on page 179](#)
- ◆ [“Directory.InfoEntry” on page 180](#)
- ◆ [“Directory.List” on page 182](#)
- ◆ [“Directory.Modify” on page 183](#)
- ◆ [“Directory.ModifyEntry” on page 184](#)
- ◆ [“Directory.ModifyGroup” on page 187](#)
- ◆ [“Directory.MoveEntries” on page 189](#)
- ◆ [“Directory.SaveAs” on page 190](#)
- ◆ [“Directory.Search” on page 191](#)
- ◆ [“Directory.ViewEntry” on page 195](#)

Directory.Accumulate.Group

Adds directory (AddressBooks) group members to the accumulator. Existing accumulator members are replaced. This action must be specified as a subcommand of [Directory.Action \(page 163\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.id (Required)

Specifies the identifier of the directory where the Directory.Item.id exists.

Directory.Item.id (Required)

Specifies the identifier of the item to be added to the accumulator.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.Accumulator.List \(page 162\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="merge" value="abaccum">
  <INPUT type="hidden" name="Directory.id" value="Frequent Contacts">
  <INPUT type="hidden" name="Directory.Item.id" Value="126">
  <INPUT type="hidden" name="Directory.Item.email" Value="Group
    Name">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT Type=submit name=Directory.Accumulate.Group value="1">
</FORM>
```

Directory.Accumulator.Add.BC

Adds an item to an accumulator as a blind carbon copy recipient. Existing accumulator members are not deleted. This action must be specified as a subcommand of [Directory.Action \(page 163\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.Item.id (Required)

Specifies the identifier of the item to be added to the accumulator.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.Accumulator.List \(page 162\)](#).

Example

```
<FORM method="post" action="/servlet/webacc">
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="merge" value="abwork">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="Directory.id" value="GroupWise Address
  Book">
  <INPUT Type=submit name=Directory.Accumulator.Add.BC value="BC">
  <INPUT Type="CheckBox" Name="Directory.Item.id"
  Value="126">JDoe.PO_1.Domain_1
</FORM>
```

Directory.Accumulator.Add.CC

Adds an item to an accumulator as a carbon copy recipient. Existing accumulator members are not deleted. This action must be specified as a subcommand of [Directory.Action](#) (page 163).

Parameters

Compose.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.Item.id (Required)

Specifies the identifier of the item to be added to the accumulator.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.Accumulator.List](#) (page 162).

Example

```
<FORM method="post" action="/servlet/webacc">
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="merge" value="abwork">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="Directory.id" value="GroupWise Address
  Book">
  <INPUT Type=submit name=Directory.Accumulator.Add.CC value="CC">
  <INPUT Type="CheckBox" Name="Directory.Item.id"
  Value="100">JDoe.PO_1.Domain_1
</FORM>
```

Directory.Accumulator.Add.To

Adds an item to an accumulator. Existing accumulator members are not deleted. This action must be specified as a subcommand of [Directory.Action](#) (page 163).

Parameters

Compose.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.Item.id (Required)

Specifies the identifier of the item to be added to the accumulator.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.Accumulator.List](#) (page 162).

Example

```
<FORM method="post" action="/servlet/webacc">
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="merge" value="abwork">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="Directory.id" value="GroupWise Address Book">
  <INPUT Type=submit name=Directory.Accumulator.Add.To value="To">
  <INPUT Type="CheckBox" Name="Directory.Item.id"
    Value="126">JDoe.PO_1.Domain_1
</FORM>
```


Directory.Accumulator.Delete

Deletes a member from the directory accumulator. This action must be specified as a subcommand of [Directory.Action](#) (page 163).

Parameters

Compose.id (Required)

Specifies the identifier of an item. This identifier is created when an item integrates with or opens the directory.

Directory.Item.recipType (Required)

Specifies the recipient type of the member being deleted: 1 (To), 2 (CC), and 3 (BC).

Directory.Item.email (Required)

Specifies the identifier of the item to be deleted from the accumulator.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.Accumulator.List](#) (page 162).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.Action
&Directory.Accumulator.Delete=&merge=abaccum&Compose.id=3
&Directory.Item.email=JDoe&Directory.Item.recipType=1"> Delete </A>
```

Directory.Accumulator.List

Lists the members in the directory accumulator. This action must be specified as subcommand of [Directory.Action](#) (page 163).

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Compose.id	Identifier of an item. This identifier is created when an item integrates with or opens the directory. For example, it can identify an email message.
Directory.id	Identifier of the directory where the specified group exists. This identifier is only used when the accumulator holds group members.
Directory.Item.comments	Comments for the group. This identifier is only used when the accumulator holds group members.
Directory.Item.count	Number of items in the accumulator.
Directory.Item.id	Identifier of the group. This identifier is only used when the accumulator holds group members.
Directory.Item.isGroup	Identifies whether the accumulator is a group. A value of 1 indicates that the accumulator is a group. This identifier is only used when the accumulator holds group members.
Directory.Item.name	Name of the group. This identifier is only used when the accumulator holds group members.
Directory.Item[n].email	Email address of the <i>n</i> th item.
Directory.Item[n].recipType	Recipient type of the <i>n</i> th item: 1 (To), 2 (CC), and 3 (BC).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.Action
&Directory.Accumulator.List=&merge=abaccum&Compose.id=3"> List </A>
```

Directory.Action

Indicates the presence of a parameter whose name is one of the Directory commands and whose value might not be present.

Parameters

One of the following Directory commands must be specified in the URL or appear as the value of the name attribute of a form object:

[Directory.Accumulate.Group \(page 157\)](#)
[Directory.Accumulator.Add.BC \(page 158\)](#)
[Directory.Accumulator.Add.CC \(page 159\)](#)
[Directory.Accumulator.Add.To \(page 160\)](#)
[Directory.Accumulator.Delete \(page 161\)](#)
[Directory.Accumulator.List \(page 162\)](#)
[Directory.CollapseGroup \(page 165\)](#)
[Directory.CopyEntries \(page 166\)](#)
[Directory.Create \(page 167\)](#)
[Directory.CreateEntry \(page 168\)](#)
[Directory.CreateGroup \(page 171\)](#)
[Directory.Delete \(page 174\)](#)
[Directory.DeleteEntry \(page 175\)](#)
[Directory.GetLastSearch \(page 176\)](#)
[Directory.HoldEntries \(page 178\)](#)
[Directory.Info \(page 179\)](#)
[Directory.InfoEntry \(page 180\)](#)
[Directory.List \(page 182\)](#)
[Directory.Modify \(page 183\)](#)
[Directory.ModifyEntry \(page 184\)](#)
[Directory.ModifyGroup \(page 187\)](#)
[Directory.MoveEntries \(page 189\)](#)
[Directory.SaveAs \(page 190\)](#)
[Directory.Search \(page 191\)](#)
[Directory.ViewEntry \(page 195\)](#)

For information about the parameters and values of a command, refer to the section for that command.

Return Values

Returns the result of the directory command used as a parameter. For information about the return value of a command, refer to the section for that command.

Example

```
<FORM method="post" action="/servlet/webacc">
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="Provider.name" value="GWAP">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="merge" value="abwork">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="Directory.id" value="GroupWise Address
  Book">
  <INPUT Type=submit name=Directory.Accumulator.Add.To value="To">
  <INPUT Type="CheckBox" Name="Directory.Item.id"
    Value="126">jdoe.PO_1.Domain_1
</FORM>
```

Directory.CollapseGroup

Cancels group edit mode. Group edit mode is initiated with [Directory.ModifyGroup \(page 187\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.Item.id (Required)

Specifies the identifier of the specified group. This identifier is only used when the accumulator holds group members.

Directory.Item.email

Specifies the email address of the group.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.Accumulator.List \(page 162\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.Action
&Directory.CollapseGroup=&merge=abaccum&Directory.Item.id="Group1
&Directory.Item.email="Jdoe,MDo"e"&Compose.id=3> Group Modify Cancel
</A>
```

Directory.CopyEntries

Copies items from a source directory to a destination directory. [Directory.HoldEntries \(page 178\)](#) is used to temporarily store the items.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.Destination.id (Required)

Specifies the identifier of the directory to which items are copied.

Directory.Source.id (Required)

Specifies the identifier of the directory from which items are copied.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.GetLastSearch \(page 176\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.CopyEntries
&Directory.Source.id=Frequent+Contacts&Directory.Destination.id=PAB
&Compose.id="5"> Copy Entry </A>
```

Directory.Create

Creates a new personal address book.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.description

Specifies the description of the directory.

Directory.name (Required)

Specifies the name of the directory.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.List \(page 182\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="Directory.Create" value="">
  <INPUT type="input" name="Directory.name" value="New PAB"><BR>
  <TEXTAREA Rows=3 Cols=50 Name="Directory.description"> New Personal
    Address Book </TEXTAREA>
</FORM>
```

Directory.CreateEntry

Creates an entry in the specified directory (personal address book). The user must have read and write access to the directory.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.id (Required)

Specifies the identifier of the directory to which the entry will be added.

Directory.Item.cellPhone

Specifies the cellular phone number associated with this entry.

Directory.Item.city

Specifies the city associated with this entry.

Directory.Item.comments

Specifies the comments associated with this entry.

Directory.Item.country

Specifies the country associated with this entry.

Directory.Item.department

Specifies the department associated with this entry.

Directory.Item.displayName

Specifies the display name associated with this entry.

Directory.Item.email

Specifies the email address for this entry. At least one of the email, firstName, or lastName parameters is required.

Directory.Item.emailDisplay

Specifies the email address associated with this entry.

Directory.Item.fax

Specifies the fax number associated with this entry.

Directory.Item.firstName

Specifies the first name or given name for this entry. At least one of the email, firstName, or lastName parameters is required.

Directory.Item.greeting

Specifies the greeting associated with this entry.

Directory.Item.homePhone

Specifies the home phone number associated with this entry.

Directory.Item.id

Specifies the identifier associated with this entry.

Directory.Item.instMessID

Specifies the instant message ID associated with this entry.

Directory.Item.lastName

Specifies the last name or surname for this entry. At least one of the email, firstName, or lastName parameters is required.

Directory.Item.mailStop

Specifies the mail stop associated with this entry.

Directory.Item.name

Specifies the name associated with this entry.

Directory.Item.orgName

Specifies the organization name associated with this entry.

Directory.Item.owner

Specifies the owner of a resource associated with this entry.

Directory.Item.pager

Specifies the pager number associated with this entry.

Directory.Item.phone

Specifies the telephone number associated with this entry.

Directory.Item.state

Specifies the state associated with this entry.

Directory.Item.street

Specifies the address associated with this entry.

Directory.Item.title

Specifies the title associated with this entry.

Directory.Item.type

Specifies the entry type:

4 Organization

5 Resource

6 Person

Directory.Item.zip

Specifies the Zip code associated with this entry.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.GetLastSearch](#) (page 176).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="action" value="Directory.CreateEntry">
  <INPUT type="hidden" name="Directory.id" value="Frequent Contacts">
  <INPUT type="input" name="Directory.Item.email"
    value="jDoe.p01.domain1">
  <INPUT type="input" name="firstName" value="John">
  <INPUT type="input" name="lastName" value="Doe">
  <INPUT type="input" name="phone" value="867-5309">
  <INPUT type="input" name="title" value="Sir">
  <INPUT type="hidden" name="Directory.Item.type" value="5">
  <INPUT type="submit" name="submit" value="Create Entry">
</FORM>
```

Directory.CreateGroup

Creates a new group. Group members can originate from a comma-delimited string of members or from the current members in the directory accumulator.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.id (Required)

Specifies the identifier of the directory where the group is saved.

Directory.Group.to

Specifies a comma-delimited string of email addresses for the To field. If Directory.Group.to, Directory.Group.cc, and Directory.Group.bc are empty, the members of the directory accumulator are used.

Directory.Group.cc

Specifies a comma-delimited string of email addresses for the CC field. If Directory.Group.to, Directory.Group.cc, and Directory.Group.bc are empty, the members of the directory accumulator are used.

Directory.Group.bc

Specifies a comma-delimited string of email addresses for the BC field. If Directory.Group.to, Directory.Group.cc, and Directory.Group.bc are empty, the members of the directory accumulator are used.

Directory.Item.comments

Specifies the comments for the new group.

Directory.Item.name (Required)

Specifies the name of the new group.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.Group.saved	The group was successfully saved.
Directory.hasMore	There are more matches than were returned.
Directory.id	Identifier for the directory that was searched.
Directory.isEmpty	1 indicates that the directory is empty.
Directory.isReadOnly	1 indicates that the directory has read-only access.
Directory.Item[n].cellPhone	Cellular phone number of the <i>n</i> th item.
Directory.Item[n].city	City of the <i>n</i> th item.

Value	Description
Directory.Item[n].comments	Comments for the <i>n</i> th item.
Directory.Item.count	Number of items returned from the search.
Directory.Item[n].country	Country of the <i>n</i> th item.
Directory.Item[n].department	Department of the <i>n</i> th item.
Directory.Item[n].displayName	Display name of the <i>n</i> th item.
Directory.Item[n].email	Email address of the <i>n</i> th item.
Directory.Item[n].emailDisplay	Displayable version of the email address of the <i>n</i> th item.
Directory.Item[n].fax	Fax number of the <i>n</i> th item.
Directory.Item[n].firstName	First name of the <i>n</i> th item.
Directory.Item[n].greeting	Greeting of the <i>n</i> th item.
Directory.Item[n].homePhone	Home phone number of the <i>n</i> th item.
Directory.Item[n].id	Identifier of the <i>n</i> th item.
Directory.Item[n].instMessID	Instant message ID of the <i>n</i> th item.
Directory.Item[n].lastName	Last name of the <i>n</i> th item.
Directory.Item[n].mailStop	Mail stop of the <i>n</i> th item.
Directory.Item[n].name	Full name of the <i>n</i> th item.
Directory.Item[n].orgName	Organization name of the <i>n</i> th item.
Directory.Item[n].owner	Owner of the <i>n</i> th item.
Directory.Item[n].pager	Pager number of the <i>n</i> th item.
Directory.Item[n].phone	Telephone number of the <i>n</i> th item.
Directory.Item[n].state	State of the <i>n</i> th item.
Directory.Item[n].street	Address of the <i>n</i> th item.
Directory.Item[n].title	Title of the <i>n</i> th item.
Directory.Item[n].type	Item type of the <i>n</i> th item: 4 Organization 5 Resource 6 Person (default)
Directory.Item[n].zip	Postal code of the <i>n</i> th item.
Directory.name	Name of the directory that was searched.

Example

```
<FORM method="post" action="/servlet/webacc">
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="Directory.CreateGroup" value="1">
  <INPUT type="hidden" name="merge" value="abgprsave">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="Directory.id" value="My Group">
  <INPUT type="input" name="Directory.name" value="My Group">
  <TEXTAREA Rows=3 Cols=50 Name="Directory.comments"> New Group
  </TEXTAREA>
</FORM>
```

Directory.Delete

Deletes an existing personal address book.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.id

Specifies the identifier of the directory to be deleted.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.List \(page 182\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.Action
&Directory.Delete=&Directory.id="New PAB"&merge=aboption&Compose.id=3">
Delete </A>
```

Directory.DeleteEntry

Deletes an entry in the specified personal address book. The user must have read and write access to the directory.

Parameters

Directory.id (Required)

Specifies the identifier of the directory from which the entry is deleted.

Directory.Item.id (Required)

Specifies the identifier of the entry to delete.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.GetLastSearch \(page 176\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.DeleteEntry
&Directory.id=Frequent+Contacts&Directory.Item.id=5"> Delete Entry
</A>
```

Directory.GetLastSearch

Retrieves the latest search result for a user. The last search performed is cached for easy and quick access.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.id (Required)

Specifies the identifier of the directory to which the entry is added.

Directory.queryCount

Specifies the maximum number of directories to retrieve.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.hasMore	Indicates whether there are more matches than were returned.
Directory.id	Identifier for the directory that was searched.
Directory.isEmpty	1 indicates that the directory is empty.
Directory.isReadOnly	1 indicates that the directory has read-only access.
Directory.Item[n].cellPhone	Cellular phone number of the <i>n</i> th item.
Directory.Item[n].city	City of the <i>n</i> th item.
Directory.Item[n].comments	Comments for the <i>n</i> th item.
Directory.Item.count	Number of items returned from the search.
Directory.Item[n].country	Country of the <i>n</i> th item.
Directory.Item[n].department	Department of the <i>n</i> th item.
Directory.Item[n].displayName	Display name of the <i>n</i> th item.
Directory.Item[n].email	Email address of the <i>n</i> th item.
Directory.Item[n].emailDisplay	Displayable version of the email address of the <i>n</i> th item.
Directory.Item[n].fax	Fax number of the <i>n</i> th item.
Directory.Item[n].firstName	First name of the <i>n</i> th item.
Directory.Item[n].greeting	Greeting of the <i>n</i> th item.
Directory.Item[n].homePhone	Home phone number of the <i>n</i> th item.
Directory.Item[n].id	Identifier of the <i>n</i> th item.

Value	Description
Directory.Item[n].instMessID	Instant message ID of the <i>n</i> th item.
Directory.Item[n].lastName	Last name of the <i>n</i> th item.
Directory.Item[n].mailStop	Mail stop of the <i>n</i> th item.
Directory.Item[n].name	Full name of the <i>n</i> th item.
Directory.Item[n].orgName	Organization name of the <i>n</i> th item.
Directory.Item[n].owner	Owner of the <i>n</i> th item.
Directory.Item[n].pager	Pager number of the <i>n</i> th item.
Directory.Item[n].phone	Telephone number of the <i>n</i> th item.
Directory.Item[n].state	State of the <i>n</i> th item.
Directory.Item[n].street	Address of the <i>n</i> th item.
Directory.Item[n].title	Title of the <i>n</i> th item.
Directory.Item[n].type	Item type of the <i>n</i> th item. Possible values are 4 (Organization), 5 (Resource), and 6 (Person). The default is 6 (Person).
Directory.Item[n].zip	Postal code of the <i>n</i> th item.
Directory.name	Name of the directory that was searched.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.GetLastSearch
&Directory.id=Frequent+Contacts&Compose.id="5"> Get Last Search </A>
```

Directory.HoldEntries

Stores directory entries temporarily. [Directory.CopyEntries \(page 166\)](#) and [Directory.MoveEntries \(page 189\)](#) use this action to store directory items. The temporary storage enables user interface changes without losing the selected items.

Parameters

Directory.id (Required)

Specifies the identifier of the directory where the items exist.

Directory.Item.id (Required)

Specifies the identifier(s) of the items to be saved.

Directory.Store.add

Specifies that new items are to be added to an existing list.

Directory.Store.removeSystemGroups

Specifies that system groups are to be removed from the stored entries.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the number of GroupWise Address Book groups that were removed from the list (`Directory.RemovedGroups.count`) and the same values as [Directory.GetLastSearch \(page 176\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.HoldEntries
&Directory.id=GroupWise Address Book&Directory.Item.id=294&Compose.id="5"> Get
Last Search </A>
```

Directory.Info

Returns the details of an existing personal address book.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.id

Specifies the identifier of the directory for which information is returned.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.description	Description of the directory.
Directory.id	Identifier of the directory.
Directory.name	Name of the directory that was searched.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.Action
&Directory.Info=&Directory.id="New PAB"&merge=aboption&Compose.id=3> Directory
Info </A>
```

Directory.InfoEntry

Retrieves information about a directory entry. A database lookup is performed on the Directory.Item.id value or the Directory.Item.firstName and Directory.Item.lastName values.

Parameters

Directory.id (Required)

Specifies the identifier of the directory where the entry is located.

Directory.Item.id (Required)

Specifies the identifier of the entry to search.

Directory.Item.firstName

Specifies the first name of the entry to search.

Directory.Item.lastName

Specifies the last name of the entry to search.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.AddEntry.saved	The entry was successfully saved.
Directory.hasMore	There are more matches than were returned.
Directory.id	Identifier for the directory that was searched.
Directory.isEmpty	1 indicates that the directory is empty.
Directory.isReadOnly	1 indicates that the directory has read-only access.
Directory.Item.cellPhone	Cellular phone number of the item.
Directory.Item.city	City of the item.
Directory.Item.comments	Comments for the item.
Directory.Item.count	Number of items returned from the search.
Directory.Item.country	Country of the item.
Directory.Item.department	Department of the item.
Directory.Item.displayName	Display name of the item.
Directory.Item.email	Email address of the item.
Directory.Item.emailDisplay	Displayable version of the email address of the item.
Directory.Item.fax	Fax number of the item.
Directory.Item.firstName	First name of the item.

Value	Description
Directory.Item.greeting	Greeting of the item.
Directory.Item.homePhone	Home phone number of the item.
Directory.Item.id	Identifier of the item.
Directory.Item.instMessID	Instant message ID of the item.
Directory.Item.lastName	Last name of the item.
Directory.Item.mailStop	Mail stop of the item.
Directory.Item.name	Full name of the item.
Directory.Item.orgName	Organization name of the item.
Directory.Item.owner	Owner of the item.
Directory.Item.pager	Pager number of the item.
Directory.Item.phone	Telephone number of the item.
Directory.Item.state	State of the item.
Directory.Item.street	Address of the item.
Directory.Item.title	Title of the item.
Directory.Item.type	Item type: 4 Organization 5 Resource 6 Person (default)
Directory.Item.zip	Postal code of the item.
Directory.name	Name of the directory that was searched.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.InfoEntry
&Directory.id=Frequent+Contacts&Directory.Item.id=5"> Entry Info </A>
```

Directory.List

Lists the GroupWise Directories (address books).

Parameters

Directory.queryCount

Specifies the maximum number of directories to retrieve.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.count	Number of directories that were retrieved.
Directory[n].id	Identifier of the <i>n</i> th directory.
Directory[n].name	Name of the <i>n</i> th directory.
Directory.isEmpty	<i>n</i> th directory is empty.
Directory.isReadOnly	<i>n</i> th directory has read only access.
Directory[n].type	Type of the <i>n</i> th item: 1 = GroupWise Address Book 7 = Frequent Contacts 10 = Personal Address Book

Example

```
<A href="/servletwebacc?User.context=xyz&action=Directory.List
&Directory.queryCount=25"> List </A>
```

Directory.Modify

Modifies the details of an existing personal address book.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.description

Specifies the description of the directory.

Directory.id (Required)

Specifies the identifier of the directory.

Directory.name (Required)

Specifies the name of the directory.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.description	The description of the directory.
Directory.id	The identifier of the directory.
Directory.name	The name of the directory.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="Directory.Modify" value="">
  <INPUT type="input" name="Directory.id" value="PAB"><BR>
  <INPUT type="input" name="Directory.name" value="PAB"><BR>
  <TEXTAREA Rows=3 Cols=50 Name="Directory.description"> Personal
    Address Book </TEXTAREA>
</FORM>
```

Directory.ModifyEntry

Modifies an entry in the specified personal directory. The user must have read and write access to the directory.

Parameters

Compose.id (Required)

The identifier of an item, as created when an item integrates with or opens the directory.

Directory.id (Required)

Specifies the identifier of the directory to which the entry will be added.

Directory.Item.cellPhone

Specifies the cellular phone number associated with this entry.

Directory.Item.city

Specifies the city associated with this entry.

Directory.Item.comments

Specifies the comments associated with this entry.

Directory.Item.country

Specifies the country associated with this entry.

Directory.Item.department

Specifies the department associated with this entry.

Directory.Item.displayName

Specifies the display name associated with this entry.

Directory.Item.email

Specifies the email address for this entry. At least one of the email, firstName, or lastName parameters is required.

Directory.Item.emailDisplay

Specifies the email address associated with this entry.

Directory.Item.fax

Specifies the fax number associated with this entry.

Directory.Item.firstName

Specifies the first name or given name for this entry. At least one of the email, firstName, or lastName parameters is required.

Directory.Item.greeting

Specifies the greeting associated with this entry.

Directory.Item.homePhone

Specifies the home phone number associated with this entry.

Directory.Item.id

Specifies the identifier associated with this entry.

Directory.Item.instMessID

Specifies the instant message ID associated with this entry.

Directory.Item.lastName

Specifies the last name or surname for this entry. The email, firstName, or lastName parameters is required.

Directory.Item.mailStop

Specifies the mail stop associated with this entry.

Directory.Item.name

Specifies the name associated with this entry.

Directory.Item.orgName

Specifies the organization name associated with this entry.

Directory.Item.owner

Specifies the owner of a resource associated with this entry.

Directory.Item.pager

Specifies the pager number associated with this entry.

Directory.Item.phone

Specifies the telephone number associated with this entry.

Directory.Item.state

Specifies the state associated with this entry.

Directory.Item.street

Specifies the address associated with this entry.

Directory.Item.title

Specifies the title associated with this entry.

Directory.Item.type

Specifies the entry type:

4 Organization

5 Resource

6 Person

Directory.Item.zip

Specifies the Zip code associated with this entry.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.GetLastSearch \(page 176\)](#).

Directory.AddEntry.saved indicates the entry was successfully saved.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="action" value="Directory.ModifyEntry">
  <INPUT type="hidden" name="Directory.id" value="Frequent Contacts">
  <INPUT type="input" name="Directory.Item.email"
    value="jDoe.p01.domain1">
  <INPUT type="input" name="firstName" value="John">
  <INPUT type="input" name="lastName" value="Doe">
  <INPUT type="input" name="phone" value="867-5309">
  <INPUT type="input" name="title" value="Sir">
  <INPUT type="hidden" name="Directory.Item.type" value="5">
  <INPUT type="submit" name="submit" value="Modify Entry">
</FORM>
```

Directory.ModifyGroup

Modifies an existing group. The group name, comments, and members can be modified.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.id (Required)

Specifies the identifier of the directory where the group is saved.

Directory.Group.to

Specifies a comma-delimited string of email addresses for the To field. If Directory.Group.to, Directory.Group.cc, and Directory.Group.bc are empty, the members of the directory accumulator are used.

Directory.Group.cc

Specifies a comma-delimited string of email addresses for the CC field. If Directory.Group.to, Directory.Group.cc, and Directory.Group.bc are empty, the members of the directory accumulator are used.

Directory.Group.bc

Specifies a comma-delimited string of email addresses for the BC field. If Directory.Group.to, Directory.Group.cc, and Directory.Group.bc are empty, the members of the directory accumulator are used.

Directory.Item.comments

Specifies the comments for the group.

Directory.Item.id (Required)

Specifies the group identifier.

Directory.Item.name (Required)

Specifies the name of the new group.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.CreateGroup \(page 171\)](#).

Example

```
<FORM method="post" action="/servlet/webacc">
  <INPUT type="hidden" name="User.context" value="rytxKjswmdAi">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="Directory.ModifyGroup" value="1">
  <INPUT type="hidden" name="merge" value="abgprsave">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="Directory.id" value="My Group">
  <INPUT type="input" name="Directory.name" value="My Modified
    Group">
  <TEXTAREA Rows=3 Cols=50 Name="Directory.comments"> Modified Group
  </TEXTAREA>
</FORM>
```

Directory.MoveEntries

Moves items from a source directory to a destination directory. [Directory.HoldEntries \(page 178\)](#) is used to temporarily store the items.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.Destination.id (Required)

Specifies the identifier of the directory to which items are copied.

Directory.Source.id (Required)

Specifies the identifier of the directory from which items are copied.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.GetLastSearch \(page 176\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.MoveEntries
&Directory.Source.id=Frequent+Contacts&Directory.Destination.id=PAB
&Compose.id="5"> Copy Entry </A>
```

Directory.SaveAs

Copies an existing personal address book and members into a new personal address book.

Parameters

Compose.id (Required)

Specifies the identifier of an item, as created when an item integrates with or opens the directory.

Directory.description

Specifies the description of the directory.

Directory.id (Required)

Specifies the identifier of the directory.

Directory.name (Required)

Specifies the name of the directory.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Directory.description	The description of the directory.
Directory.id	The identifier of the directory.
Directory.name	The name of the directory.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="Compose.id" value="3">
  <INPUT type="hidden" name="action" value="Directory.Action">
  <INPUT type="hidden" name="Directory.SaveAs" value="">
  <INPUT type="input" name="Directory.id" value="PAB-SaveAs"><BR>
  <INPUT type="input" name="Directory.name" value="PAB-SavaAs"><BR>
  <TEXTAREA Rows=3 Cols=50 Name="Directory.description"> Personal
    Address Book-SaveAs </TEXTAREA>
</FORM>
```

Directory.Search

Searches a directory for specified entries.

Parameters

Directory.id (Required)

Specifies the identifier of the directory to search.

Directory.queryCount

Specifies the number of directory items to retrieve.

Directory.Query[n].field

Specifies which field in the row is searched in the directory.

Valid search fields in the GroupWise Address Book are:

- ◆ Directory.Item.department
- ◆ Directory.Item.firstName
- ◆ Directory.Item.lastName
- ◆ Directory.Item.name

Valid search fields in a Personal Address Book are:

- ◆ Directory.Item.cellPhone
- ◆ Directory.Item.city
- ◆ Directory.Item.country
- ◆ Directory.Item.department
- ◆ Directory.Item.displayName
- ◆ Directory.Item.email
- ◆ Directory.Item.fax
- ◆ Directory.Item.firstName
- ◆ Directory.Item.greeting
- ◆ Directory.Item.homePhone
- ◆ Directory.Item.instMessID
- ◆ Directory.Item.lastName
- ◆ Directory.Item.mailStop
- ◆ Directory.Item.name
- ◆ Directory.Item.orgName
- ◆ Directory.Item.owner
- ◆ Directory.Item.pager
- ◆ Directory.Item.phone
- ◆ Directory.Item.state
- ◆ Directory.Item.street
- ◆ Directory.Item.title
- ◆ Directory.Item.zip

Directory.Query[n].filter

Specifies the filter type that is used in the search. Valid filter types are

- ◆ Directory.Query.Filter.begin
- ◆ Directory.Query.Filter.equal
- ◆ Directory.Query.Filter.notBegin
- ◆ Directory.Query.Filter.nEqual

Directory.Query[n].operator

Specifies an operation that will be used in the search. Valid operations are

- ◆ Directory.Query.Operator.and
- ◆ Directory.Query.Operator.end
- ◆ Directory.Query.Operator.not
- ◆ Directory.Query.Operator.or

Directory.Query[n].value

Specifies a unique value to search for in this row.

Directory.Query.itemType

Specifies the item type to be searched. Valid types are

- ◆ Directory.Item.Type.all
- ◆ Directory.Item.Type.group
- ◆ Directory.Item.Type.organization
- ◆ Directory.Item.Type.person
- ◆ Directory.Item.Type.resource

Directory.queryFirst

Specifies the first name to use for the query. This parameter exists for backward compatibility only.

Directory.queryLast

Specifies the last name to use for the query. This parameter exists for backward compatibility only.

Directory.queryType

Specifies whether to query from the beginning of the list or from the current position. Valid values are

- ◆ AddressBook.queryFirst(Default)
- ◆ AddressBook.queryNext

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Return Value	Description
Directory.count	Number of directories retrieved.

Return Value	Description
Directory.hasMore	There are more matches than were returned.
Directory[n].id	Identifier for the <i>n</i> th directory.
Directory[n].isEmpty	1 indicates that the <i>n</i> th directory is empty.
Directory[n].isReadOnly	1 indicates that the <i>n</i> th directory has read-only access.
Directory.Item[n].cellPhone	Cellular phone number of the <i>n</i> th item.
Directory.Item[n].city	City of the <i>n</i> th item.
Directory.Item[n].comments	Comments for the <i>n</i> th item.
Directory.Item.count	Number of items returned from the search.
Directory.Item[n].country	Country of the <i>n</i> th item.
Directory.Item[n].department	Department of the <i>n</i> th item.
Directory.Item[n].displayName	Display name of the <i>n</i> th item.
Directory.Item[n].email	Email address of the <i>n</i> th item.
Directory.Item[n].emailDisplay	Displayable version of the email address of the <i>n</i> th item.
Directory.Item[n].fax	Fax number of the <i>n</i> th item.
Directory.Item[n].firstName	First name of the <i>n</i> th item.
Directory.Item[n].greeting	Greeting of the <i>n</i> th item.
Directory.Item[n].homePhone	Home phone number of the <i>n</i> th item.
Directory.Item[n].id	Identifier of the <i>n</i> th item.
Directory.Item[n].instMessID	Instant message ID of the <i>n</i> th item.
Directory.Item[n].lastName	Last name of the <i>n</i> th item.
Directory.Item[n].mailStop	Mail stop of the <i>n</i> th item.
Directory.Item[n].name	Full name of the <i>n</i> th item.
Directory.Item[n].orgName	Organization name of the <i>n</i> th item.
Directory.Item[n].owner	Owner of the <i>n</i> th item.
Directory.Item[n].pager	Pager number of the <i>n</i> th item.
Directory.Item[n].phone	Telephone number of the <i>n</i> th item.
Directory.Item[n].state	State of the <i>n</i> th item.
Directory.Item[n].street	Address of the <i>n</i> th item.
Directory.Item[n].title	Title of the <i>n</i> th item.
Directory.Item[n].type	Item type of the <i>n</i> th item: 4 Organization 5 Resource 6 Person (default)
Directory.Item[n].zip	Postal code of the <i>n</i> th item.

Return Value	Description
Directory[n].name	Name of the <i>n</i> th directory.
Directory.queryCount	Number of directory items to retrieve.
Directory.queryFirst	First name.
Directory.queryLast	Last name.
Directory.queryString	Query string used to perform the latest search.
Directory.queryTypeFiler	Directory.Query.itemType value used to perform the latest search.
Directory[n].type	1 GroupWise Address Book 7 Frequent Contacts (PAB) 10 Personal Address Book

Remarks

A search is composed of several parameters. A valid search has the following parameters:

- ◆ Directory.queryCount—Defined one time for each search.
- ◆ Directory.id—Defined one time for each search.
- ◆ Directory.Query.itemType—Defined one time for each search.

You must also provide a query string that can have 1 or more rows. A query row has the following parameters:

- ◆ Directory.Query[n].field
- ◆ Directory.Query[n].filter
- ◆ Directory.Query[n].value
- ◆ Directory.Query[n].operator

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="action" value="Directory.Search">
  <INPUT type="hidden" name=" Directory.queryCount" value="25">
  <INPUT type="hidden" name="Directory.id" value="PAB">
  <INPUT type="hidden" name="Directory.Query.itemType" value="
    Directory.Item.Type.person">
  <INPUT type="input" name="Directory.Query.1.field"
    value="Directory.Item.firstName">
  <INPUT type="input" name="Directory.Query.1.filter"
    value="Directory.Query.Filter.betin">
  <INPUT type="input" name="Directory.Query.1.value" value="John">
  <INPUT type="input" name="Directory.Query.1.operator"
    value="Directory.Query.Operation.and">
  <INPUT type="submit" name="submit" value="Search">
</FORM>
```

Directory.ViewEntry

Retrieves information about a directory entry. A database lookup is not performed. Instead, information is retrieved from the cache.

Parameters

Directory.id (Required)

Specifies the identifier of the directory where the entry is located.

Directory.Item.id (Required)

Specifies the identifier of the entry to search.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.InfoEntry \(page 180\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.ViewEntry
&Directory.id=Frequent+Contacts&Directory.Item.id=5"> Entry Info </A>
```

10.5 Document

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Document.Profile” on page 197](#)
- ◆ [“Document.Save” on page 198](#)
- ◆ [“Document.View” on page 199](#)
- ◆ [“Document.ViewNative” on page 200](#)

Document.Profile

Retrieves the profile information for the specified document.

Parameters

Library.id (Required)

Specifies the library identifier of the desired document.

Document.id (Required)

Specifies the document number of the desired document.

Document.version (Required)

Specifies the numeric version number of the desired version of the document, or one of the following values:

Official
Current

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Library.id	Library identifier for the document.
Document.id	Document number for the document.
Document.version	Document version.
Document.author	Author of the document.
Document.size	Size of the document.
Document.subject	Document subject.
Document.extension	Document file extension (such as doc).
Document.Profile.createDate	Creation date of the document.
Document.Profile.modifyDate	Modification date of the document.
Document.Profile.officialVersion	Version number for the official document.
Document.Profile.currentVersion	Version number for the current document.
Document.Profile.versionCount	Number of versions for the document.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=
Document.Profile&Document.Library.id=Library1&Document.id=3452
&Document.version=Official">Document Properties</A>
```

Document.Save

Downloads the specified document for saving.

Parameters

Provider.name

Specifies the name of the GroupWise WebAccess provider, GWAP.

Item.type

Specifies the document Reference type, DREF.

Library.id

Specifies the identifier of the document library.

Document.id

Specifies the document number.

Document.version

Specifies the document number, or one of the following values:

Official

Current

Document.format

Specifies GWNative.

Document.section

Specifies GWContentRoot.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the original contents of the document.

Remarks

Saving a document from a document reference requires the parameters to be specified in the extra path info of the URL. This is the portion of the URL that follows the name of the servlet and occurs before the question mark.

The parameters need to occur in a specific order:

```
/servlet/webacc/User.context/Provider/Item.type/Document.Library.Id/Document.id/  
Download.version/Document.format/Document.section
```

Example

```
<A href="/servlet/webacc/xyz/GWAP/DREF/Lib1/6542/Official/GWNative/  
GWContentRoot?action=Document.Save"> Save </A>
```

Document.View

Downloads a document for viewing.

Parameters

Document.View is invoked in a similar manner as [Document.Save \(page 198\)](#).

Return Values

Returns an HTML version of the document, if a conversion is possible.

Example

```
<A href="/servlet/webacc/xyz/GWAP/DREF/Lib1/6542/Official/GWNative/  
GWContentRoot?action=Document.View"> View Document </A>
```

Document.ViewNative

Downloads a document for viewing with a local browser plug-in or helper application.

Parameters

Document.ViewNative is invoked in a similar manner as [Document.Save \(page 198\)](#).

Return Values

Returns a version of the document in its native format. If an appropriate viewer exists on the user's machine, the document is displayed in that viewer. Otherwise, the user is asked to save the file to disk.

Example

```
<A href="/servlet/webacc/xyz/GWAP/DREF/Lib1/6542/Official/GWNative/  
GWContentRoot?action=Document.ViewNative"> View Document </A>
```


10.6 Find

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Find.Action” on page 202](#)
- ◆ [“Find.Beginning” on page 203](#)
- ◆ [“Find.First” on page 204](#)
- ◆ [“Find.NewUpdate” on page 208](#)
- ◆ [“Find.Next” on page 209](#)
- ◆ [“Find.Prev” on page 210](#)
- ◆ [“Find.StopSearch” on page 211](#)
- ◆ [“Find.Update” on page 212](#)

Find.Action

Performs the specified action.

Parameters

One of the following Find commands must be specified in the URL:

[Find.Beginning \(page 203\)](#)

[Find.First \(page 204\)](#)

[Find.NewUpdate \(page 208\)](#)

[Find.Next \(page 209\)](#)

[Find.Prev \(page 210\)](#)

[Find.StopSearch \(page 211\)](#)

[Find.Update \(page 212\)](#)

The existence of one of these values in the URL determines which action is invoked. Refer to the documentation of each command for its parameter information.

Return Values

For specific return values, see the documentation for each command.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.First" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="Radio" Name="Find.bQuerySubject" CHECKED
    value="false">
  <INPUT NAME="Find.queryText" TYPE="text" VALUE="test" SIZE=25 >
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

Find.Beginning

Displays the first specified number of items found from a search that was initiated by [Find.First \(page 204\)](#). This action does not issue a new search. The results are retrieved from the cached results. Find.Beginning must be specified as a subcommand of [Find.Action \(page 202\)](#) and must be preceded by Find.First.

Parameters

Find.Beginning (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use Find.UpdateList for templates that use frames and use Find.UpdateCache for all other templates.

Folder.queryCount

Specifies the number of items to display.

Return Values

Returns the same values as [Find.First \(page 204\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.Beginning" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

Find.First

Initiates a search into the user's message store using the search parameters provided. Find.First must be specified as a subcommand of [Find.Action](#) (page 202).

Parameters

Find.First (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use Find.UpdateList for templates that use frames and use Find.UpdateCache for all other templates.

Find.enableRefresh

Specifies to enable automatic refreshing of the message list while searching.

Find.refreshRate

Specifies the number of seconds to wait before automatically refreshing the page.

Find.queryText

Specifies the text to search for.

Find.bQuerySubject

Specifies whether to search for text string in message subject:

true Search for query text only in message subject

false Search for query text anywhere in message

Find.queryName

Specifies the text to search for in address fields.

Find.bQueryToCC

Specifies whether to search for queryName in To/From fields only:

true Search for queryName only in To/From fields

false Search for queryName only in CC/BC fields

Find.folders

Specifies a comma-delimited list of folder IDs that were selected in the folder list. If a collapsed folder is checked, the sub folders do not need to be identified and returned.

Find.folderDrns

Specifies a comma-delimited list of folder DRNs selected in the folder list. This parameter can be used as an alternative to the Find.folders parameter.

Find.Sourt.draft

Specifies to search only draft message items. This parameter can be combined with other Find.Source parameters.

Find.Type.mail

Specifies to search only mail message items. This parameter can be combined with other Find.Type parameters.

Find.Type.note

Specifies to search only note message items. This parameter can be combined with other Find.Type parameters.

Find.Type.task

Specifies to search only task message items. This parameter can be combined with other Find.Type parameters.

Find.Type.appointment

Specifies to search only appointment message items. This parameter can be combined with other Find.Type parameters.

Find.Type.phone

Specifies to search only phone message items. This parameter can be combined with other Find.Type parameters.

Find.Type.document

Specifies to search only document items. This parameter can be combined with other Find.Type parameters.

Find.Source.received

Specifies to search only received message items. This parameter can be combined with other Find.Source parameters.

Find.Source.sent

Specifies to search only sent message items. This parameter can be combined with other Find.Source parameters.

Find.Source.posted

Specifies to search only posted message items. This parameter can be combined with other Find.Source parameters.

Return Values

Value	Description
Find.UpdateList or Find.UpdateCache	Defines the find list window as folder content.
Find.enableRefresh	Set if the page is to be automatically refreshed while searching.
Find.refreshRate	Number of seconds to wait before automatically refreshing the page.
Find.bSearching	1 specifies the server is still searching. The find results are not complete.
Folder.hasMore	1 specifies there are more messages than what was returned.
Folder.hasInbound	1 specifies the returned list of items contains received items.
Folder.hasOutbound	1 specifies the returned list of items contains send items.

Value	Description
Folder.name	Name of the folder.
Folder.queryCount	Display this number of items in the folder.
Folder.enablePrev	1 specifies that the previous button should be displayed.
Folder.enableNext	1 specifies that the display next button should be displayed.
Find.resultsTotal	Total number of items found. Might not be complete if bSearching is set.
Find.resultsFirstDispltem	Relative to the complete find results list, the number of first item displayed.
Find.resultsLastDispltem	Relative to the complete find results list, the number of last item displayed.
Item.count	Number of items returned.
Item[n].id	Identifier of the <i>n</i> th item within the list of items returned.
Item[n].drn	GroupWise specific identifier of the <i>n</i> th item returned.
Item[n].type	Type of the <i>n</i> th item: Appointment Document Mail Note Phone Task
Item[n].priority	Priority of the <i>n</i> th item.
Item[n].from	Sender of the <i>n</i> th item.
Item[n].told	List of user identifiers to whom the <i>n</i> th item was sent.
Item[n].subject	Subject of the <i>n</i> th item.
Item[n].subType	Subtype of the <i>n</i> th item: Item.SHARED_ADDRESS_BOOK Item.PERSONAL Item.SHARED_FOLDER Item.DISCUSSION Item.WORKFLOW Item.REPLY_REQUESTED
Item[n].hasAttachment	Indicates that the <i>n</i> th item has an attachment.
Item[n].folder	Name of the folder that contains the <i>n</i> th item.
Item[n].dateSent	Send date and time of the <i>n</i> th item.
Item[n].dateStart	Start date and time of the <i>n</i> th item.
Item[n].dateEnd	End date and time of the <i>n</i> th item.
Item[n].Status.x	Status of the <i>n</i> th item, where x is any of the following: accepted completed read

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPg">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.First" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="Radio" Name="Find.bQuerySubject" CHECKED
    value="false">
  <INPUT NAME="Find.queryText" TYPE="text" VALUE="test" SIZE=25 >
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

Find.NewUpdate

Launches a new search using the query parameters established with [Find.First \(page 204\)](#). This action must be specified as a subcommand of [Find.Action \(page 202\)](#). It must also be preceded by a [Find.First](#) action.

Parameters

Find.NewUpdate (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use [Find.UpdateList](#) for templates that use frames and use [Find.UpdateCache](#) for all other templates.

Return Values

Returns the same values as [Find.First \(page 204\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.NewUpdate" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```


Find.Next

Displays the next number of specified items found from a search that was initiated by [Find.First \(page 204\)](#). Find.Next must be specified as a subcommand of [Find.Action \(page 202\)](#) and preceded by Find.First.

Parameters

Find.Next (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use Find.UpdateList for templates that use frames and use Find.UpdateCache for all other templates.

Folder.queryCount

Specifies the number of items to display.

Return Values

Returns the same values as [Find.First \(page 204\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.Next" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

Find.Prev

Displays the previous number of specified items found from a search that was initiated by [Find.First \(page 204\)](#). Find.Prev must be specified as a subcommand of [Find.Action \(page 202\)](#) and preceded by Find.First.

Parameters

Find.Prev (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use Find.UpdateList for templates that use frames and use Find.UpdateCache for all other templates.

Folder.queryCount

Specifies the number of items to display.

Return Values

Returns the same values as [Find.First \(page 204\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.Prev" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

Find.StopSearch

Aborts a search that was initialized by [Find.First \(page 204\)](#). Find.bSearching can be used to determine if the server is still searching. This action must be specified as a subcommand of [Find.Action \(page 202\)](#). It must also be preceded by a Find.First action.

Parameters

Find.StopSearch (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use Find.UpdateList for templates that use frames and use Find.UpdateCache for all other templates.

Return Values

None. Can be used with Find.Update Sub-Action to return the latest results.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.StopSearch" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

Find.Update

Updates the currently displayed range of items found in a search that was initiated with [Find.First \(page 204\)](#). Find.Update can also be used to update the display of items found as the server is continuing to search. Find.bSearching can be used to determine if the server is still searching. This must be specified as a subcommand of [Find.Action \(page 202\)](#). It must also be preceded by a Find.First action.

Parameters

Find.Update (Required)

Specifies the required subaction.

User.context (Required)

Specifies the context identifier for this user's session.

Find.UpdateList or Find.UpdateCache (Required)

Specifies the folder list context to the find window. Use Find.UpdateList for templates that use frames and use Find.UpdateCache for all other templates.

Folder.queryCount

Specifies the number of items to display.

Return Values

Returns the same values as [Find.First \(page 204\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="psquRmoverPq">
  <INPUT type="hidden" name="action" value="Find.Action">
  <INPUT type="hidden" name="Find.Update" value="true">
  <INPUT type="hidden" name="Find.UpdateList" value="1">
  <INPUT type="hidden" name="merge" value="msglist">
</FORM>
```

10.7 Folder

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Directory.Cancel” on page 214](#)
- ◆ [“Directory.GetRecipients” on page 215](#)
- ◆ [“Folder.Action” on page 217](#)
- ◆ [“Folder.Create” on page 218](#)
- ◆ [“Folder.Delete” on page 219](#)
- ◆ [“Folder.Directory” on page 220](#)
- ◆ [“Folder.Empty” on page 221](#)
- ◆ [“Folder.Expand” on page 222](#)
- ◆ [“Folder.List” on page 223](#)
- ◆ [“Folder.ListChild” on page 225](#)
- ◆ [“Folder.ListShare” on page 227](#)
- ◆ [“Folder.Open” on page 228](#)
- ◆ [“Folder.Share” on page 231](#)

Directory.Cancel

Cancels the GroupWise Address Book that was opened with [Folder.Directory \(page 220\)](#). The shared folder list is not modified. This action must be specified as a subcommand of [Folder.Action \(page 217\)](#).

Parameters

Directory.Cancel (Required)

Specifies the required subaction.

Compose.id (Required)

Specifies the identifier of the shared folder list in the GroupWise Address Book, as created by [Folder.Directory \(page 220\)](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Folder.ListShare \(page 227\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Folder.Action
&Directory.Cancel=&Compose.id=5"> Close Directory </A>
```

Directory.GetRecipients

Populates the TO, CC, and BC fields in a rule with the list of selected users from the GroupWise Address Book.

Parameters

Directory.GetRecipients (Required)

Specifies the required subaction.

Compose.id (Required)

Specifies the identifier of the rule in the GroupWise Address Book, as created with [Rule.Directory \(page 304\)](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns to the rule template. Information that has been added to the rule fields will be returned.

Value	Description
Folder.id	Move items to this folder.
Folder.drn	Move items to this folder.
Item.bc	List of BC recipients.
Item.cc	List of CC recipients.
Item.message	Message text.
Item.subject	Message subject.
Item.to	List of TO recipients.
RuleCondition	Type of condition: Contains, Does Not Contain, Begins With, Matches.
RuleConditionfield	GroupWise field used for rule comparison: From, Message, Subject, To.
Rule.Condition.text	Comparison text for the rule condition.
Rule.drn	Unique identifier of the rule.
Rule.name	Rule name.
Rule.replyTo	Reply To mode: 0 Reply to Sender and 1 Reply to All.
Rule.subjectPrefix	Prefix for the subject field: Fwd:, Re:.
Rule.type	Rule type: Delete, Forward, Move to Folder, Reply, and Vacation.

Remarks

Directory.GetRecipients is necessary only if you have used any of the following Directory Accumulator actions.

- ♦ [Directory.Accumulator.Add.BC \(page 158\)](#)
- ♦ [Directory.Accumulator.Add.CC \(page 159\)](#)
- ♦ [Directory.Accumulator.Add.To \(page 160\)](#)
- ♦ [Directory.Accumulator.Delete \(page 161\)](#)
- ♦ [Directory.Accumulator.List \(page 162\)](#)

Directory.GetRecipients must be specified as a subcommand of [Rule.Action \(page 298\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Rule.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT type=image name=Directory.GetRecipients src="btokus.gif">
</FORM>
```


Folder.Action

Performs a specified action.

Parameters

One of the following Folder commands must be specified in the URL or appear as the name of a form object.

[Directory.Cancel \(page 214\)](#)

[Directory.GetRecipients \(page 297\)](#)

[Folder.Create \(page 218\)](#)

[Folder.Delete \(page 219\)](#)

[Folder.Directory \(page 220\)](#)

[Folder.Empty \(page 221\)](#)

[Folder.Expand \(page 222\)](#)

[Folder.ListShare \(page 227\)](#)

[Folder.Share \(page 231\)](#)

The existence of one of these values in the URL determines the action that is invoked. For information about the parameters of a command, refer to the section for that command.

Return Values

Returns the result of the folder command used as a parameter. For information about the return value of a command, refer to the section for that command.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Folder.Action>
  <INPUT type=hidden name=Folder.Share value=1>
  <INPUT type=hidden name=Folder.id value=10>
</FORM>
```

Folder.Create

Creates a personal folder.

Parameters

Folder.accept

Specifies if you are accepting a shared folder, which acceptance makes this parameter required.

Folder.name (Required)

Specifies the name of the folder.

Folder.position (Required)

Specifies where in the folder list the new folder should be inserted:

Folder.Sibling.n	Required. Specifies that the new folder is a sibling to the <i>n</i> th folder. (The new folder appears in the folder list immediately after the <i>n</i> th folder.)
Folder.Child.n	Required. Specifies that the new folder is a child to the <i>n</i> th folder.
Folder.SiblingDrn.n	Required. Specifies that the new folder is a sibling to the folder with this DRN. (The new folder appears in the folder list immediately after the specified folder.) This value can be used as an alternative to the Folder.Sibling value.
Folder.ChildDrn.n	Required. Specifies that the new folder is a child to the folder with this DRN. This value can be used as an alternative to the Folder.Child value.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Folder.List \(page 223\)](#).

Example

```
/servlet/webacc?User.context=xyz&action=Folder.Create&
Folder.name=New+Folder&Folder.position=Folder.Sibling.5
```

Folder.Delete

Deletes the specified folder.

Parameter

Folder.id or Folder.drn (Required)

Specifies the ID or the DRN of the folder to be deleted.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Folder.List \(page 223\)](#).

Example

```
/servlet/webacc?User.context=xyz&action=Folder.Delete &Folder.id=6
```

Folder.Directory

Integrates shared folders with the GroupWise Address Book and opens the GroupWise Address Book. Folder.SaveShare is required to save the list of accumulated users. This action must be specified as a subcommand of the [Folder.Action \(page 217\)](#).

Parameters

In addition to passing the parameters used by [Directory.List \(page 182\)](#), it might be helpful to pass the Compose.id into the process (template) doing the directory so that it will be available for adding recipients to the directory.

Compose.id

Specifies the identifier of the shared folder item, as returned from Folder.Directory.

Folder.Directory (Required)

Specifies the required action.

User context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Directory.List \(page 182\)](#).

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Folder.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT type=submit name=Folder.Directory value="Address Book">
</FORM>
```

Folder.Empty

Deletes all of the items in the specified folder (for a Trash folder only).

Parameters

Folder.id or Folder.drn (Required)

Specifies the ID of the DRN of the Trash folder.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same items as [Folder.Open \(page 228\)](#). If Folder.count is 0, there are no items.

Example

```
/servlet/webacc? User.context=xyz&action=Folder.Empty& Folder.id=1
```

Folder.Expand

Toggles the expanded state of the specified folder to show or hide subfolders.

Parameters

Folder.id or Folder.drn (Required)

Specifies the ID or the DRN of the folder to expand or collapse.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Folder.List \(page 223\)](#).

Example

```
/servlet/webacc?User.context=xyz&action=Folder.Expand& Folder.id=1
```

Folder.List

Retrieves a list of folders.

Parameters

Folder.expandAll

Specifies to display all folders.

User.context (Required)

Specifies the context identifier for this user's session.

Find.Folders.clear

Specifies to clear all selected folders (used with the Find feature) if true.

Return Values

Value	Description
Folder.count	Number of folders returned.
Folder[n].id	Folder ID of the <i>n</i> th folder.
Folder[n].drn	Folder DRN of the <i>n</i> th folder.
Folder[n].name	Name of the <i>n</i> th folder.
Folder[n].level	Level of the <i>n</i> th folder. A child folder has a level that is one greater than its parent's level.
Folder[n].parentId	Folder ID of the parent for the <i>n</i> th folder.
Folder[n].type	Type of the <i>n</i> th folder: Folder.NORMAL Folder.SENTITEMS Folder.TASKLIST Folder.QUERY Folder.USER Folder.UNIVERSAL Folder.TRASH Folder.CABINET Folder.SHARED Folder.UNOPENED Folder.CHECKLIST Folder.DOCUMENT Special backward compatibility case with GroupWise 5.2. Folder.SHAREDREF If this is a shared folder reference, display the user's rights to the folder: Folder[n].Rights.delete Delete rights to the <i>n</i> th folder. Folder[n].Rights.post Post or create rights to the <i>n</i> th folder.
Folder[n].inRule	1 specifies the <i>n</i> th folder is referenced within a rule.

Value	Description
Folder[n].hasSubfolders	1 specifies the <i>n</i> th folder has subfolders.
Folder[n].isExpanded	1 specifies the <i>n</i> th folder is expanded and its children are included in the folder list.

Example

```
servlet/webacc?User.contet=xyz&action=Folder.List
```


Folder.ListChild

Retrieves child folders for the specified folder. The child folders are not expanded.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Folder.id or Folder.drn (Required)

Specifies the folder identifier or Folder DRN of the parent folder. Only the unexpanded child folders are returned.

Return Values

Value	Description
Folder.count	Number of folders returned.
Folder[n].id	Folder ID of the <i>n</i> th folder.
Folder[n].drn	Folder DRN of the <i>n</i> th returned folder.
Folder[n].name	Name of the <i>n</i> th folder.
Folder[n].level	Level of the <i>n</i> th folder. A child folder has a level that is one greater than its parent's level.
Folder[n].parentId	Folder ID of the parent for the <i>n</i> th level.
Folder[n].type	Specifies the type of the <i>n</i> th item: Folder.NORMAL Folder.SENTITEMS Folder.TASKLIST Folder.QUERY Folder.USER Folder.UNIVERSAL Folder.TRASH Folder.CABINET Folder.SHARED Folder.UNOPENED Folder.CHECKLIST Folder.DOCUMENT Special backward compatibility case with GroupWise 5.2. Folder.SHAREDREF If this is a shared folder reference, display the user's rights to the folder. Folder[n].Rights.delete Delete rights to the <i>n</i> th folder. Folder[n].Rights.post Post or create rights to the <i>n</i> th folder.
Folder[n].inRule	1 specifies the <i>n</i> th folder is referenced within a rule.
Folder[n].hasSubfolders	1 specifies the <i>n</i> th folder has subfolders.

Value	Description
Folder[n].isExpanded	1 specifies the <i>n</i> th folder is expanded. Its children are included in the folder list.

Example

```
<A href="/servletwebacc?User.context=xyz&action=Folder.ListChild&Folder.id=10"> List Child </A>
```

Folder.ListShare

Retrieves a list of users who have rights to a shared folder.

Parameters

Folder.id or Folder.drn (Required)

Specifies the folder identifier or Folder DRN.

Folder.ListShare (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Folder.id	Folder ID.
Folder.drn	Folder DRN.
Folder.name	Name of the folder.
Folder.Share.User.count	Shared with this number of users.
Folder.Share.User[n].email	Email address of the user.
Folder.Share.User[n].name	Name of the user.
Folder.Share.User[n].updName	User.PostOffice.Domain name.
Folder.Share.User[n].Rights.add	Right to create in the shared folder.
Folder.Share.User[n].Rights.delete	Right to delete in the shared folder.
Folder.Share.User[n].Rights.edit	Right to edit the shared folder.
Folder.Share.User[n].Rights.read	Right to read the shared folder.
Folder.Share.User[n].Rights.owner	Owner rights to the shared folder.
Folder.Share.User[n].Status.accepted	User accepted the shared folder.
Folder.Share.User[n].Status.deleted	User deleted the shared folder.
Folder.Share.User[n].Status.owner	Owner of the shared folder.
Folder.Share.User[n].Status.pending	User has not acted on the shared folder.
Folder.Share.User[n].Status.read	User has opened the shared folder.
Folder.type	Type of folder: Folder.SHARED.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Folder.Action
&Folder.ListShare=&Folder.id=5"> List Shared Folder </A>
```

Folder.Open

Retrieves the message list for the specified folder.

Parameters

Folder.id or Folder.drn

Specifies the identifier or DRN of the folder to open. The last accessed folder is used if this parameter is not present.

Folder.queryCount

Specifies how many items to retrieve.

Folder.queryType

Specifies whether to query from the beginning of the list or the current position. If this parameter is not specified, the query starts at the beginning of the list.

Item.queryFirst Begin item list with the first item in the folder (Default)

Item.queryNext Begin folder list where the last list of folders items ended

User.context (Required)

Specifies the context identifier for this user's session.

Item.Accept.ShowAs

Specifies one of the following acceptance levels for an appointment:

- 1 Busy
- 2 Free
- 4 Out of office
- 8 Tentative

Return Values

Value	Description
Folder.hasMore	1 specifies there are more messages than what was returned.
Folder.hasDraft	1 specifies that the returned list of items contains one or more draft items.
Folder.id	ID of the folder.
Folder.drn	DRN of the folder.
Folder.name	Name of the folder.
Folder.level	Level of the folder. A child folder has a level that is one level greater than its parent's level.
Folder.parentId	Folder ID of the parent folder.
Folder.queryCount	Display this number of items in the folder.

Value	Description
Folder.type	Type of the <i>n</i> th folder: Folder.NORMAL Folder.SENTITEMS Folder.TASKLIST Folder.QUERY Folder.USER Folder.UNIVERSAL Folder.TRASH Folder.CABINET Folder.SHARED Folder.CHECKLIST Folder.UNOPENED Folder.DOCUMENT Special backward compatibility with GroupWise 5.2. Folder.SHAREDREF If this is a shared folder reference, display the user's rights to the folder. Folder[n].Rights.delete Delete rights to the <i>n</i> th folder. Folder[n].Rights.post Post or create rights to the <i>n</i> th folder.
Folder.inRule	1 indicates the <i>n</i> th folder is referenced within a rule.
Folder.hasSubfolders	1 indicates the <i>n</i> th folder has subfolders.
Folder.isExpanded	1 indicates the <i>n</i> th folder is expanded. Its children are included in the folder list.
Item.count	Number of items returned.
Item[n].id	Identifier of the <i>n</i> th item within the list of returned items.
Item[n].drn	GroupWise specific identifier of the <i>n</i> th returned item.
Item[n].type	Type of the <i>n</i> th item: Appointment, Note, or Task.
Item[n].priority	Priority of the <i>n</i> th item.
Item[n].from	Sender of the <i>n</i> th item.
Item[n].told	List of user identifiers to whom the <i>n</i> th item was sent.
Item[n].subject	Subject of the <i>n</i> th item.
Item[n].subType	Subtype of the <i>n</i> th item: Item.SHARED_ADDRESS_BOOK Item.PERSONAL Item.SHARED_FOLDER Item.DISCUSSION Item.WORKFLOW Item.REPLY_REQUESTED
Item[n].Folder.newName	Name given to a shared folder when it is being accepted.
Item[n].Status.read	Specifies read or unread.
Item[n].Status.accepted	Specifies accepted or unaccepted.

Value	Description
Item[n].Status.completed	Specifies completed or uncompleted.
Item[n].dateSent	Date the <i>n</i> th item was sent.
Item[n].dateStart	Start time for the <i>n</i> th item for calendar items.
Item[n].dateEnd	End time for the <i>n</i> th item for appointments, or due date for tasks.
Item[n].location	Location of the <i>n</i> th item for appointments.
Item[n].isOutbound	The <i>n</i> th item is a sent item.
Item[n].Priority.task	Priority of the <i>n</i> th item.
Item[n].dateTaskAssigned	Assigned date for the <i>n</i> th item for tasks.
Item[n].Library.id	Library ID of the <i>n</i> th item for document references.
Item[n].Document.id	Document ID of the <i>n</i> th item for document references.
Item[n].Document.version	Document version of the <i>n</i> th item for document references: Official, Current, or numeric version number.
Item[n].Document.ext	Document extension of the <i>n</i> th item for document references.
Item[n].hasAttachment	Specifies if the <i>n</i> th item has attachments.
Folder.hasInbound	1 specifies that the returned list of items contains received items.
Folder.hasOutbound	1 specifies that the returned list of items contains send items.
User.Access.HTMLDoc	1 enables viewing documents in HTML format (global).
User.Access.HTMLDocNative	1 enables viewing documents in native format (global).
Item[n].allowViewNative	1 enables native viewing for this document.
Item[n].allowView	1 enables HTML viewing for this document.

Example

```
/servlet/webacc?User.context=xyz&action=Folder.Open&Folder.id=2
```

Folder.Share

Modifies a shared folder user list. The shared folder user list is not stored on the server. Instead, the user list, with respective user rights, is passed to the server with each request to add, delete, or save the shared list.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Folder.Share (Required)

Specifies the required action.

Folder.id or Folder.drn (Required)

Specifies the folder identifier.

Folder.SaveShare (SUBACTION)

Specifies to save the shared folder user list.

Folder.Share.subject

Specifies the subject of the shared folder.

Folder.Share.User.email

Specifies to add the specified user to the shared folder list.

ADD.<email address>

Specifies to grant create rights in a shared folder.

DELETE.<email address>

Specifies to grant delete rights in a shared folder.

DELETEUSER.<email address>

Specifies to delete the specified user from the shared folder list.

EDIT.<email address>

Specifies to grant edit rights in a shared folder.

READ.<email address>

Specifies to grant read rights in a shared folder.

Return Values

Returns the same values as [Folder.ListShare \(page 227\)](#).

Saving Folder Lists

Specifying the Folder.SaveShare subaction saves the shared folder list.

Adding Users

Specifying the Folder.Share.User.email subaction with a value of the user name or email address adds a user to the shared folder list. Alternatively, the GroupWise Address Book (Folder.Directory action) can be used to look up multiple users that can be added to the shared folder list.

Deleting Users

Specifying the DELETEUSER.[email or name] removes a user from the shared folder list.

Each user can have four different rights to a shared folder: add, delete, edit, and read. These specific rights are preappended to the user's email address and passed to the server during a request.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Folder.Action>
  <INPUT type=hidden name=Folder.id value=1>
  <INPUT type=checkbox name=ADD.jDoe.po.domain value=on>
  <INPUT type=checkbox name=READ.jDoe.po.domain value=on>
  <INPUT type=image name=Folder.SaveShare src="btsaveus.gif">
</FORM>
```


10.8 Library

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Library.List” on page 234](#)
- ◆ [“Library.Search” on page 235](#)

Library.List

Lists the available document libraries.

Parameters

Library.queryCount

Specifies the number of libraries to return.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Library.count	Number of document libraries.
Library.Default.id	Identifier for the user's default library.
Library[n].id	Identifier for the <i>n</i> th document library.
Library[n].name	Name of the <i>n</i> th document library.
Library[n].description	Description of the <i>n</i> th document library.

Example

```
/servlet/webacc?User.context=xyz&action=Library.List
```

Library.Search

Searches the specified document library.

Parameters

Library.queryField

Specifies which fields to search:

ALL - Search all fields (Default)

AUTHOR - Search author fields

SUBJECT - Search subject fields

Library.queryId

Specifies the name of the library to search. This parameter can be repeated to search as many libraries as desired.

Library.queryCount

Specifies how many documents to retrieve.

Library.queryType

Specifies whether to query from the beginning of the list or the current position:

Document.queryFirst Return results starting with the first document found (Default)

Document.queryNext Return results starting where the last set of results ended

Library.queryText

Specifies the text to use for the query. If not present or blank, defaults to find all documents.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Library.queryCount	Number of documents requested.
Library.queryField	Search field used.
Library.queryText	Search criteria.
Library.queryId	Libraries searched.
Library.hasMore	1 specifies that more documents were found than what was returned.
Document.count	Number of documents found.
Document[n].Library.id	Library identifier of the <i>n</i> th document.
Document[n].id	Document number of the <i>n</i> th document.
Document[n].version	Document version of the <i>n</i> th document.
Document[n].author	Author of the <i>n</i> th document.

Value	Description
Document[n].size	Size of the <i>n</i> th document.
Document[n].subject	Subject of the <i>n</i> th document.
Document[n].extension	Document file extension of the <i>n</i> th document.
Document[n].Profile.createDate	Creation date of the <i>n</i> th document.
Document[n].Profile.modifyDate	Modification date of the <i>n</i> th document.
Document[n].Profile.officialVersion	Official version of the <i>n</i> th document.
Document[n].Profile.currentVersion	Current version of the <i>n</i> th document.
Document[n].Profile.versionCount	Number of versions for the <i>n</i> th document.
User.Access.HTMLDoc	1 enables viewing documents in HTML format (global)
User.Access.HTMLDocNative	1 enables viewing documents in native format (global)
Document[n].allowViewNative	1 enables native viewing for this attachment
Document[n].allowView	1 enables HTML viewing for this attachment

Example

```
<A href=/servlet/webacc?User.context=xyz&action=Library.Search&Library.queryField=ALL&Library.queryId=Lib1&Library.queryId=Lib2&Library.queryText=WebAccess+Documents"> WebAccess Documents </A>
```

10.9 Item

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Item.Accept” on page 238](#)
- ◆ [“Item.AcceptAll” on page 239](#)
- ◆ [“Item.AcceptSharedAddressBook” on page 240](#)
- ◆ [“Item.Action” on page 241](#)
- ◆ [“Item.CheckList.Bottom” on page 243](#)
- ◆ [“Item.CheckList.Down” on page 244](#)
- ◆ [“Item.CheckList.Save” on page 245](#)
- ◆ [“Item.CheckList.Top” on page 246](#)
- ◆ [“Item.CheckList.Up” on page 247](#)
- ◆ [“Item.Complete” on page 248](#)
- ◆ [“Item.Compose” on page 249](#)
- ◆ [“Item.Decline” on page 250](#)
- ◆ [“Item.DeclineAll” on page 251](#)
- ◆ [“Item.Delete” on page 252](#)
- ◆ [“Item.MarkRead” on page 253](#)
- ◆ [“Item.Move” on page 254](#)
- ◆ [“Item.Next” on page 255](#)
- ◆ [“Item.Previous” on page 256](#)
- ◆ [“Item.Properties” on page 257](#)
- ◆ [“Item.Properties.Read” on page 259](#)
- ◆ [“Item.Properties.ReadInfo” on page 262](#)
- ◆ [“Item.Properties.ReadNext” on page 265](#)
- ◆ [“Item.Purge” on page 266](#)
- ◆ [“Item.Read” on page 267](#)
- ◆ [“Item.ReadChild” on page 271](#)
- ◆ [“Item.ReadLater” on page 272](#)
- ◆ [“Item.ReadParent” on page 273](#)
- ◆ [“Item.Retract” on page 274](#)
- ◆ [“Item.RetractAll” on page 275](#)
- ◆ [“Item.Undelete” on page 276](#)

Item.Accept

Accepts the specified item (applies only to appointments, personal address books, and tasks).

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to accept (can occur more than once so that multiple items can be accepted at the same time).

Item.comment

Specifies the comment to include with the acceptance of the item.

User.context (Required)

Specifies the context identifier for this user's session.

Item.Accept.ShowAs

Specifies one of the following acceptance levels for an appointment:

- 1 Busy
- 2 Free
- 4 Out of office
- 8 Tentative

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Accept>
  <INPUT type=checkbox name=Item.drn value==342z3z>
  <INPUT type=checkbox name=Item.drn value==582z4z>
</FORM>
```

Item.AcceptAll

Accepts all occurrences of an auto-date appointment or task.

Remarks

The parameters and return values are the same as [Item.Accept \(page 238\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.AcceptAll&Item.drn=533z8z">  
Accept All Instances </A>
```

Item.AcceptSharedAddressBook

Accepts the specified shared address book.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to accept. This parameter can occur more than once so that multiple items can be accepted at the same time.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If this action was specified as a subaction of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, returns the same values as [Folder.Open \(page 228\)](#). Otherwise, returns nothing.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.AcceptSharedAddressBook>
  <INPUT type=checkbox name=Item.drn value==342z3z>
  <INPUT type=checkbox name=Item.drn value==582z4z>
</FORM>
```


Item.Action

Performs the specified action on items.

Parameters

One of the following subactions must be specified:

[Folder.List \(page 223\)](#)
[Item.Accept \(page 238\)](#)
[Item.AcceptAll \(page 239\)](#)
[Item.CheckList.Bottom \(page 243\)](#)
[Item.CheckList.Down \(page 244\)](#)
[Item.CheckList.Save \(page 245\)](#)
[Item.CheckList.Top \(page 246\)](#)
[Item.CheckList.Up \(page 247\)](#)
[Item.Complete \(page 248\)](#)
[Item.Decline \(page 250\)](#)
[Item.DeclineAll \(page 251\)](#)
[Item.Delete \(page 252\)](#)
[Item.MarkRead \(page 253\)](#)
[Item.Move \(page 254\)](#)
[Item.Next \(page 255\)](#)
[Item.Previous \(page 256\)](#)
[Item.Properties \(page 257\)](#)
[Item.Properties.Read \(page 259\)](#)
[Item.Properties.ReadInfo \(page 262\)](#)
[Item.Properties.ReadNext \(page 265\)](#)
[Item.Purge \(page 266\)](#)
[Item.Read \(page 267\)](#)
[Item.ReadChild \(page 271\)](#)
[Item.ReadLater \(page 272\)](#)
[Item.ReadParent \(page 273\)](#)
[Item.Retract \(page 274\)](#)
[Item.RetractAll \(page 275\)](#)
[Item.Undelete \(page 276\)](#)

The existence of one of these values in the URL determines the action that is invoked on the items. Refer to the documentation for each command for parameter information. In addition to the parameters for the subaction, the following parameters can be specified:

Item.UpdateList

Specifies how to update the item list:

`Calendar.Search` Update the list by doing a calendar search. For the other parameters that must accompany this option, see [Calendar.Search \(page 110\)](#).

For any other value, the item list is updated with the items in the last access folder, which is the equivalent of [Folder.Open \(page 228\)](#).

Return Values

See the documentation for each action for the pertinent return information.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=checkbox name=Item.drn value==582z1z>Task 1
  <INPUT type=checkbox name=Item.drn value==236z2z>Task 2
  <INPUT type=submit name=Item.Decline value="Decline Task">
</FORM>
```

Item.CheckList.Bottom

Moves the selected items to the bottom of the checklist folder.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item or items to move to the bottom of the checklist folder.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If this action was specified as a subaction of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=hidden name="Item.CheckList.Bottom" value="">
  <INPUT type=checkbox name=Item.drn value=582z1z>
  <INPUT type=hidden name=Item.UpdateList value=1>
</FORM>
```

Item.CheckList.Down

Moves the selected items down in the checklist folder.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item or items to move down the checklist folder.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If this action was specified as a subaction of [Item.Action \(page 241\)](#) and [Item.UpdateList](#) was specified, returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=hidden name="Item.CheckList.Down" value="">
  <INPUT type=checkbox name=Item.drn value=582z1z>
  <INPUT type=hidden name=Item.UpdateList value=1>
</FORM>
```

Item.CheckList.Save

Saves the checklist information for an item. Valid checklist information includes the completed status and due date.

Parameters

Calendar.queryDay

Specifies the desired day of the month.

Calendar.queryMonth

Specifies the desired month of the year (1=January, 12=December).

Calendar.queryYear

Specifies the desired year (in 4-digit format, such as 1999 or 2000).

Item.CheckList.completed

Specifies if the completed box is enabled (if the check box is selected).

Item.CheckList.dueDate

Specifies if the due date is enabled (if the check box is selected).

Item.drn (Required)

Specifies the GroupWise identifier of the item.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Item.Read \(page 267\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=hidden name="Item.CheckList.Save" value="">
  <INPUT type=hidden name=Item.drn value=582z1z>
  <INPUT type=checkbox name=Item.CheckList.completed value="enabled">
  <INPUT type=text name=Calendar.queryDay value="8">
  <INPUT type=text name=Calendar.queryMonth value="7">
  <INPUT type=text name=Calendar.queryYear value="2002">
  <INPUT type=hidden name=Item.CheckList.completed value="enabled">
</FORM>
```

Item.CheckList.Top

Moves the selected items to the top of the checklist folder.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item or items to move to the top of the checklist folder.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If this action was specified as a subaction of [Item.Action \(page 241\)](#) and [Item.UpdateList](#) was specified, returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=hidden name="Item.CheckList.Top" value="">
  <INPUT type=checkbox name=Item.drn value=582z1z>
  <INPUT type=hidden name=Item.UpdateList value=1>
</FORM>
```

Item.CheckList.Up

Moves the selected items up in the checklist folder.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item or item to move up the checklist folder.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If this action was specified as a subaction of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=hidden name="Item.CheckList.Up" value="">
  <INPUT type=checkbox name=Item.drn value=582z1z>
  <INPUT type=hidden name=Item.UpdateList value=1>
</FORM>
```

Item.Complete

Marks the specified task as complete. This action is valid only for task or workflow items (Item.subType = Item.WORKFLOW).

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the task to mark complete. This parameter can occur multiple times to complete several items with one request.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns Item.action.folderUpdate to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and Item.UpdateList was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=checkbox name=Item.drn value==582z1z>Task 1
  <INPUT type=checkbox name=Item.drn value==236z2z>Task 2
  <INPUT type=submit name=Item.Accept value="Accept Task">
  <INPUT type=submit name=Item.Complete value="Complete Task">
</FORM>
```


Item.Compose

Reads the specified draft item and sets the compose mode.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the draft item to open. This method has no effect if the item specified is not a draft item.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Item.Read \(page 267\)](#).

Item.Dcline

Declines the specified appointments and tasks.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to accept. This parameter can occur more than once so that multiple items can be accepted at the same time.

Item.comment

Specifies the comment to include with the acceptance of the item.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=checkbox name=Item.drn value==582z1z>Task 1
  <INPUT type=checkbox name=Item.drn value==236z2z>Task 2
  <INPUT type=submit name=Item.Accept value="Accept Task">
  <INPUT type=submit name=Item.Dcline value="Decline Task">
</FORM>
```

Item.DeclineAll

Declines the specified auto-date appointments and tasks.

Remarks

The parameters and return values are the same as for [Item.Decline](#) (page 250).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.DeclineAll&Item.drn=533z8z">  
Decline All Instances </A>
```

Item.Delete

Deletes the specified item.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the message to delete. This parameter can occur multiple times to delete several items with the same transaction.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=checkbox name=Item.drn value==582z1z>Task 1
  <INPUT type=checkbox name=Item.drn value==236z2z>Task 2
  <INPUT type=submit name=Item.Accept value="Accept Task">
  <INPUT type=submit name=Item.Delete value="Delete Items">
</FORM>
```

Item.MarkRead

Marks an item as having been read.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to be marked. So that several items can be marked with a single transaction, this parameter can occur multiple times.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Item.MarkRead returns Item.action.folderUpdate to indicate that the folder list view might need to be updated.

If Item.MarkRead was specified as a submethod of [Item.Action \(page 241\)](#) and Item.UpdateList was specified, Item.MarkRead returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.ActionRead&Item.MarkRead=&Item.drn=533z8z"> Mark Item Read </A>
```

Item.Move

Moves the specified items.

Parameters

Folder.id or Folder.drn (Required)

Specifies the destination folder.

Item.drn (Required)

Specifies the GroupWise identifier of the item to be moved. This parameter can occur multiple times to move several items with the same transaction.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If an attempt was made to move a draft item to an invalid folder, Folder.Move.Err.folder (which is set to the name of the destination folder) is returned.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and Item.UpdateList was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=submit name="Folder.id" value="10">
  <INPUT type=submit name="Item.drn" value="544">
  <INPUT type=submit name=Item.Move value="1">
</FORM>
```

Item.Next

Opens the next item in the message list.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to open.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Item.Read \(page 267\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=submit name="Item.drn" value="544">
  <INPUT type=submit name="Item.Next" value="1">
</FORM>
```

Item.Previous

Opens the previous item in the message list.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the current item.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Item.Read \(page 267\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=submit name="Item.drn" value="544">
  <INPUT type=submit name="Item.Previous" value="1">
</FORM>
```


Item.Properties

Retrieves the properties for the specified message. Part of the properties include the recipients and their respective status information.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Item.Properties[n].name	Name of recipient.
Item.Properties[n].id	
Item.Properties[n].read	
Item.Properties[n].Read.time	
Item.Properties[n].accepted	
Item.Properties[n].Accepted.time	
Item.Properties[n].Accepted.comment	
Item.Properties[n].completed	
Item.Properties[n].Completed.time	
Item.Properties[n].declined	
Item.Properties[n].Declined.time	
Item.Properties[n].Declined.comment	
Item.Properties[n].delegated	
Item.Properties[n].Delegated.time	
Item.Properties[n].Delegated.comment	
Item.Properties[n].deleted	
Item.Properties[n].Deleted.time	
Item.Properties[n].delivered	
Item.Properties[n].Delivered.time	
Item.Properties[n].downloaded	
Item.Properties[n].Downloaded.time	
Item.Properties.moreRecipients	Set if there are more recipients to display.

Value	Description
Item.Properties.boxType	Valid values: 1 The user viewing the item is a recipient. Don't show item status information or BC users. 2 The user viewing the item is the owner. Show item status information and BC users.
Item.Properties[n].copyType	Valid values: 1 Recipient in the To field. 2 Recipient in the CC field. 3 Recipient in the BC field.
Item.Properties[n].typePostOffice	1 if the item is a post office instead of a recipient.
Item.Properties[n].purged	
Item.Properties[n].Purged.time	
Item.Properties[n].replied	
Item.Properties[n].Replied.time	
Item.Properties[n].retracted	
Item.Properties[n].Retracted.time	
Item.Properties[n].started	
Item.Properties[n].Started.time	
Item.Properties.Total.count	Number of recipients and post offices in the list.
Item.Properties[n].transferred	
Item.Properties[n].Transferred.time	
Item.Properties[n].undeleted	
Item.Properties[n].Undeleted.time	
Item.Properties[n].undeliverable	
Item.Properties.Users.count	Number of recipients in the list.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.properties&Item.drn=434z6z">
Item Info </A>
```

Item.Properties.Read

Reads property information regarding the selected item. Currently, this returns the property information for a post office.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to open.

Item.Properties.id (Required)

Specifies the location in the list of the selected item.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Item.Properties.name	Name of recipient.
Item.Properties.id	Identifier of item.
Item.Properties.copyType	Valid Values: 1 Recipient in the To field 2 Recipient in the CC field 3 Recipient in the BC field
Item.Properties.GWDomain.name	Name of the GroupWise domain.
Item.Properties.InternetDomainName	Name of the Internet domain.
Item.Properties.accepted	
Item.Properties.Accepted.time	
Item.Properties.Accepted.comment	
Item.Properties.boxType	Valid Values: 1 The user viewing the item is a recipient. Don't show item status information or BC users. 2 The user viewing the item is the owner. Show item status information and BC users.
Item.Properties.completed	
Item.Properties.Completed.time	
Item.Properties.declined	
Item.Properties.Declined.time	
Item.Properties.Declined.comment	
Item.Properties.delegated	

Value	Description
Item.Properties.Delegated.time	
Item.Properties.Delegated.comment	
Item.Properties.Deleted.time	
Item.Properties.delivered	
Item.Properties.Delivered.time	
Item.Properties.downloaded	
Item.Properties.Downloaded.time	
Item.Properties.Forwarded	
Item.Properties.Forwarded.time	
Item.Properties.Host.deleted	
Item.Properties.Host.Deleted.time	
Item.Properties.Incomplete	
Item.Properties.Incomplete.time	
Item.Properties.purged	
Item.Properties.Purged.time	
Item.Properties.read	
Item.Properties.Read.time	
Item.Properties.replied	
Item.Properties.Replied.time	
Item.Properties.retracted	
Item.Properties.Retracted.time	
Item.Properties.started	
Item.Properties.Started.time	
Item.Properties.transferred	
Item.Properties.Transferred.time	
Item.Properties.Transfer.delayed	
Item.Properties.TransferDelayed.time	
Item.Properties.Transfer.failed	
Item.Properties.TransferFailed.reason	
Item.Properties.TransferFailed.time	
Item.Properties.Transfer.pending	
Item.Properties.Transfer.pending	
Item.Properties.TransferPending.time	

Value	Description
Item.Properties.typePostOffice	1 if the item is of type post office (instead of a recipient).
Item.Properties.undeleted	
Item.Properties.Undeleted.time	
Item.Properties.undeliverable	
Item.Properties.Undeliverable.users	
Item.Properties.UndeliverableUser.time	

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.Action&Item.properties.Read=&Item.drn=434z6z">Item Selected Item Info </A>
```

Item.Properties.ReadInfo

Reads other property information about an item, such as send options, attachments, and other relevant information.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to open.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Item.Properties.drn	Identifier of item.
Item.Properties.type	Valid Values: 1 Mail 2 Note 4 Task 8 Appointment 16 Phone
Item.Properties.boxType	Valid Values: 1 The user viewing the item is a recipient. Don't show item status information or BC users. 2 The user viewing the item is the owner. Show item status information and BC users.
Item.Properties.fromText	Sender of the item.
Item.Properties.subject	Subject of the item.
Item.Properties.linUserHash	User information (used for problem solving).
Item.Properties.createDate	Creation date of the item.
Item.Properties.expirationDate	Expiration date of the item.
Item.Properties.distributionDate	Distribution date of the item.
Item.Properties.scheduledDate	Scheduled date of the item.
Item.Properties.security	Security settings on the item: 1 Normal 2 Proprietary 3 Confidential 4 Secret 5 Top Secret 6 For your eyes only

Value	Description
Item.Properties.priority	Item priority: Low Medium High
Item.Properties.autoDelete	Yes or No.
Item.Properties.concealSubject	Yes or No.
Item.Properties.notifyRecipient	Yes or No.
Item.Properties.ReplyRequested.type	Type of reply requested: WhenConvenient or ByDate.
Item.Properties.ReplyRequested.day	Number of days selected for a reply request.
Item.Properties.ReturnNotification.accepted	Type of return notification for an accepted item: 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.Properties.ReturnNotification.completed	Type of return notification for a completed item: 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.Properties.ReturnNotification.deleted	Type of return notification for a deleted item: 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.Properties.ReturnNotification.declined	Type of return notification for a declined item: 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.Properties.ReturnNotification.opened	Type of return notification for an opened item: 1 None 2 Mail 4 Notify 8 Notify and Mail
Item.Properties.statusTracking	Type of status tracking: 1 Delivered 2 Delivered and Opened 3 All Information
Item.Properties.Attachment.count	Number of files attached.

Value	Description
Item.Properties.Attachment[n].lastModifiedDate	Last modified date for the selected attachment.
Item.Properties.Attachment[n].name	Attachment name.
Item.Properties.Attachment.size	Attachment size.
Item.Properties.Attachment[n].type	Valid attachment types are: File Folder Message Sound View Appointment Note Task Mail Form Multimedia Phone Ole Profile (Document)

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.Action&Item.properties.ReadInfo=&Item.drn=434z6z"> Get Other Properties </A>
```


Item.Properties.ReadNext

Retrieves the next set of recipients (if `Item.Properties.moreRecipients` is defined).

Parameters

Same as [Item.Properties](#) (page 257).

Example

```
A href="/servlet/  
webacc?User.context=xyz&action=Item.Action&Item.properties.ReadNext=&Item.drn=434z  
6z"> Get More Recipients</A>
```

Item.Purge

Purges the specified item in the Trash folder.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to be purged. This parameter can occur multiple times so that several items can be purged with a single transaction.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Item.Action>
  <INPUT type=checkbox name=Item.drn value==582z1z>Task 1
  <INPUT type=submit name=Item.Purge value="1">
</FORM>
```

Item.Read

Reads the specified item.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to be read.

User.context (Required)

Specifies the context identifier for this user's session.

Item.Accept.ShowAs

Specifies one of the following acceptance levels for an appointment:

- 1 Busy
- 2 Free
- 4 Out of office
- 8 Tentative

NOTE: This parameter is used with GroupWise 7 SP 1 and later versions only.

Return Values

Value	Description
Item.Attachment[n].allowViewNative	1 enables native viewing for this document or attachment.
Item.Attachment[n].allowView	1 enables HTML viewing for this document or attachment.
Item.bc	List of BC recipients.
Item.bcName	Display names of the BC recipients of the enclosed item.
Item.caller	Caller of the enclosed phone item.
Item.cc	List of CC recipients.
Item.cclId	User's IDs of the CC recipients of the enclosed item.
Item.ccName	Display names of the To recipients of the enclosed item.
Item.cFullId	Full resolved IDs of the CC recipients of the enclosed item.
Item.CheckList.dueDate	CheckList due date of the selected item.
Item.Child.depth	Depth of enclosed item. Indicates how deeply nested the enclosure is. For example, if you're forwarding a forwarded item, the original item would have a depth of 2.
Item.Child.id	Enclosed ID of the item. This item is an enclosure such as a forwarded or delegated item.
Item.company	Company of the enclosed phone item.
Item.dateEnd	End date or Due date of the enclosed item.

Value	Description
Item.dateSent	Send date of the enclosed item.
Item.dateStart	Start date of the enclosed item.
Item.drn	GroupWise specific identifier for item.
Item.duration	Duration if the item is an appointment.
Item.durationType	Duration type for appointments: 1 Minutes 2 Hours 3 Days
Item.Folder.newName	Used when accepting a shared folder.
Item.from	Sender of the enclosed item.
Item.fromFullId	Full ID (userid.postoffice.domain) of the sender of the enclosed item.
Item.id	Identifier of item.
Item.isAutoDate	1 specifies that the enclosed item is an auto-date item.
Item.isDraft	Specifies the item is a draft item.
Item.isOutbound	1 specifies that the item is a sent item.
Item.location	Location of the enclosed appointment item.
Item.message	Message of the enclosed item.
Item.phone	Phone of the enclosed phone item.
Item.priority	Priority of the enclosed item.
Item.Reply	(Optional) Included if the Item.Read is the result of a reply request: sender (reply only to sender) or all (reply to all).
Item.ReplyRequested.date	Date associated with the reply request of the enclosed item.
Item.ReplyRequested.day	During a compose, this value is the number of days before a reply is requested.
Item.ReplyRequested.type	Type of reply requested of the enclosed item: WhenConvenient or ByDate.
Item.security	Security of the enclosed item.
Item.Status.accepted	Accepted status of the enclosed item: Accepted or Unaccepted.
Item.Staus.completed	Completed status of the enclosed item: Completed or Uncompleted.
Item.Status.isDeleted	Deleted status of an item. 1 indicates that the item has been deleted.
Item.Status.read	Read status of the enclosed item: Read or Unread.
Item.subject	Subject of the enclosed item.
Item.subType	Sub-type of the enclosed item.
Item.Task.priority	Priority of the enclosed item.
Item.Task.startDate	Start date of the enclosed task item.

Value	Description
Item.to	List of To recipients.
Item.toFullId	Full resolved IDs of the To recipients of the enclosed item.
Item.toName	Display names of the To recipients of the enclosed item.
Item.told	User's IDs of the To recipients of the enclosed item.
Item.type	Item type: Mail, Appointment, Note, Task, or Phone.
Signature.isEnabled	A signature is enabled or disabled.
Signature.isAutomatic	A signature can be automatically added or the user can be prompted to add a signature to outgoing messages.
Signature.signature	Signature text.
User.Access.HTMLDoc	1 enables viewing documents in HTML format (global).
User.Access.HTMLDocNative	1 enables viewing documents in native format (global).
User.Access.View	1 enables viewing attachments in HTML format (global).
User.Access.ViewNative	1 enables viewing attachments in native format (global).

If the message is being posted to a folder, add some information about the folder:

Value	Description
Folder.id	Identifier of folder.
Folder.drn	Folder DRN of the folder.
Folder.name	Name of the folder.
Folder.type	Specifies the type of the <i>n</i> th folder: Folder.NORMAL Folder.SENTITEMS Folder.TASKLIST Folder.QUERY Folder.USER Folder.UNIVERSAL Folder.TRASH Folder.CABINET Folder.SHARED Folder.CHECKLIST Folder.UNOPENED Folder.DOCUMENT Special backward compatibility case with GroupWise 5.2. Folder.SHAREDREF If this is a shared folder reference, it displays the user's rights to the folder.
Folder[n].Rights.delete	Delete rights to the <i>n</i> th folder.
Folder[n].Rights.post	Post or create rights to the <i>n</i> th folder.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.Read&Item.drn=434z6z"> Read  
Item </A>
```

Item.ReadChild

Reads a child item, such as an enclosed, forwarded message.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the current item.

Item.Attachment.id (Required)

Specifies the attachment ID of the child item to read.

Item.Child.id (Required)

Specifies the child ID of the item to read. It is possible for the child ID to contain spaces, so you must URL encode this value. For example:

```
{VAR encodeURIComponent(Item.Child.id)}
```

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Item.Read \(page 267\)](#).

Example

```
<A href="/servletwebacc?User.context=xyz&action=Item.ReadChild&Item.Attachment.id=1&Item.Child.id={VAR encodeURIComponent(Item.Child.id)}&Item.drn=434z6z">Read Fowarded Message</A>
```

Item.ReadLater

Marks an item as unread.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to be marked unread. This parameter can occur multiple times so that several items can be marked unread with a single transaction.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.Action
&Item.ReadLater=&Item.drn=533z8z"> Read Item Later </A>
```


Item.ReadParent

Reads a child item, such as an enclosed, forwarded message.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the current item.

Item.Child.depth (REQUIRED if greater than 1)

Specifies the depth of the current item and is necessary if greater than 1.

Item.Child.id (Required)

Specifies the child ID of the item to be read. It is possible for the child ID to contain spaces, so you must URL encode this value. For example,

```
{VAR urlEncode(Item.Child.id)}
```

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Item.Read \(page 267\)](#).

Example

```
{! ----- Close enclosed message -----  
! This closes the reading of a message that is being forwarded or delegated.  
! (It was opened from clicking its link on the compose page.)  
! -----}  
<FORM Method=Post  
  Action="/servlet/webacc">  
  <INPUT type="hidden" name="User.context" value="{VAR User.context}">  
  <INPUT type="hidden" name="action" value="Item.Action">  
  {IF Item.Child.depth > 1}  
    <INPUT type="hidden" name="Item.Child.id" value="{VAR Item.Child.id}">  
    <INPUT type="hidden" name="Item.Child.depth" value="{VAR  
      Item.Child.depth}">  
  {ENDIF}  
  <INPUT type="hidden" name="Item.drn" value="{VAR Item.drn}">  
  <INPUT type="hidden" name="merge" value="msgitem">  
  {IF Item.Child.depth > 1}  
    <INPUT type="hidden" name="Url.envelope" value="1">  
  {ENDIF}  
  {IF {STRING Button.type} == "IMAGE"}  
    <INPUT Type=image Name="Item.ReadParent"  
      alt="{STRING Button.Image.AltText.btnCancel}"  
      src="{VAR Templates.Images.url}/btnclose{VAR User.lang}.gif"  
      border=0>  
  {ELSE}  
    <INPUT TYPE="Submit" Name="Item.ReadParent"  
      Value=" {STRING Button.Text.btnCancel} ">  
  {ENDIF}  
</FORM>
```

Item.Retract

Retracts the specified sent items.

Parameters

Item.comment

Specifies a comment to include with the retraction of an item.

Item.drn (Required)

Specifies the GroupWise identifier of the item to be purged. This parameter can occur multiple times so that several items can be purged with a single transaction.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Item.RetractAll

Retracts the specified sent item.

Parameter

Item.comment

Specifies the comment to include with the retraction of an item.

Item.drn (Required)

Specifies the GroupWise identifier of the item to be purged (can occur multiple times so that several items can be purged with a single transaction).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<A href="/servlet webacc?User.context=xyz&action=Item.RetractAll&Item.drn=533z8z"> Retract from all Mailboxes </A>
```

Item.Undelete

Undeletes the specified item from the Trash.

Parameters

Item.drn (Required)

Specifies the GroupWise identifier of the item to be purged. This parameter can occur multiple times so that several items can be purged with a single transaction.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

As a result of the action performed, this method returns `Item.action.folderUpdate` to indicate that the folder list view might need to be updated.

If this method was specified as a submethod of [Item.Action \(page 241\)](#) and `Item.UpdateList` was specified, it returns the same values as [Folder.Open \(page 228\)](#). Otherwise, nothing else is returned.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Item.Undelete&Item.drn=533z8z">
Undelete </A>
```

10.10 Miscellaneous

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“XKI.getDRNList” on page 278](#)

XKI.getDRNList

Retrieves a list of message item DRNs. You can specify new, modified, or all items. The request can, optionally, specify a folder to which to restrict the DRN list.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

XKI.getDRNList (Required)

Specifies the required action.

XKI.getDRNList.type (Required)

Specifies the type of list to return:

getDRNList.new	Returns all new items after XKI.getDRNList.startTime
getDRNList.modified	Returns all modified items after XKI.getDRNList.startTime
getDRNList.all	Returns all items.

XKI.getDRNList.startTime (Required for getDRNList.new and getDRNList.modified)

Specifies the start time (the number of seconds since 12:00:00 a.m. on January 1, 1970) from which to search for new and modified items.

Folder.drn (Optional)

Specifies the identifier or DRN of the folder to retrieve the DRN list from. Trash, Query, or Contacts folders cannot be specified. The Sent Items folder cannot be specified with getDRNList.new.

Return Values

XKI.Item.count	Total number of items in the list.
XKI.Item[n].drn	GroupWise specific identifier of the <i>n</i> th returned item.

Remarks

You cannot access XKI.getDRNList as a proxy user.

Also, shared folders are not included when querying across all folders.

Example

```
/servlet/webacc?User.context=xyz&action=XKI.getDRNList&XKI.getDRNList.type=XKI.getDRNList.modified&XKI.getDRNList.startTime=1091170800
```

10.11 Proxy

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Directory.Cancel” on page 280](#)
- ◆ [“Directory.GetRecipients” on page 281](#)
- ◆ [“Proxy.Action” on page 282](#)
- ◆ [“Proxy.AddAccess” on page 283](#)
- ◆ [“Proxy.CloseListAccess” on page 285](#)
- ◆ [“Proxy.CloseListHistory” on page 286](#)
- ◆ [“Proxy.DeleteAccess” on page 287](#)
- ◆ [“Proxy.DeleteHistory” on page 288](#)
- ◆ [“Proxy.Directory” on page 289](#)
- ◆ [“Proxy.ListAccess” on page 290](#)
- ◆ [“Proxy.ListHistory” on page 291](#)
- ◆ [“Proxy.Login” on page 292](#)
- ◆ [“Proxy.Modify Access” on page 293](#)
- ◆ [“Proxy.ReadAccess” on page 295](#)

Directory.Cancel

Cancels the GroupWise Address Book that was opened with [Proxy.Directory \(page 289\)](#). This action must be specified as a subcommand of [Proxy.Action \(page 282\)](#). The proxy list is not modified.

Parameters

Directory.Cancel (Required)

Specifies the required subcommand.

Compose.id (Required)

Specifies the identifier of the proxy list in the GroupWise Address Book, as created with [Proxy.Directory \(page 289\)](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

If modifying the proxy access list, returns the same values as [Proxy.ListAccess \(page 290\)](#).

If modifying the proxy history list, returns the same values as [Proxy.ListHistory \(page 291\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action
&Directory.Cancel=&Compose.id=5"> Close Directory </A>
```


Directory.GetRecipients

Populates the proxy access or history list with the list of selected users from the GroupWise Address Book and closes the GroupWise Address Book.

Parameters

Directory.GetRecipients (Required)

Specifies the required subaction.

Compose.id (Required)

Specifies the identifier of the proxy access or history list in the GroupWise Address Book, as created with [Proxy.Directory](#) (page 289).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the selected User.Id and any proxy rights selected before opening the directory.

Remarks

Directory.GetRecipients is necessary only if you have used any of the following Directory Accumulator actions:

- ♦ [Directory.Accumulator.Add.BC](#) (page 158)
- ♦ [Directory.Accumulator.Add.CC](#) (page 159)
- ♦ [Directory.Accumulator.Add.To](#) (page 160)
- ♦ [Directory.Accumulator.Delete](#) (page 161)
- ♦ [Directory.Accumulator.List](#) (page 162)

Directory.GetRecipients must be specified as a subcommand of [Proxy.Action](#) (page 282).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Proxy.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT type=image name=Directory.GetRecipients src="btokus.gif">
</FORM>
```

Proxy.Action

Performs a specified action.

Parameters

One of the following Proxy commands must be specified in the URL or appear as the name of a form object:

[Directory.Cancel \(page 280\)](#)
[Directory.GetRecipients \(page 281\)](#)
[Proxy.AddAccess \(page 283\)](#)
[Proxy.CloseListAccess \(page 285\)](#)
[Proxy.CloseListHistory \(page 286\)](#)
[Proxy.DeleteAccess \(page 287\)](#)
[Proxy.DeleteHistory \(page 288\)](#)
[Proxy.Directory \(page 289\)](#)
[Proxy.ListAccess \(page 290\)](#)
[Proxy.ListHistory \(page 291\)](#)
[Proxy.Login \(page 292\)](#)
[Proxy.ModifyAccess \(page 293\)](#)
[Proxy.ReadAccess \(page 295\)](#)

The existence of one of these values in the URL determines the action that is invoked. For information about the parameters and values of a command, refer to the section for that command.

Return Values

Returns the result of the proxy command used as a parameter. For information about the return value of a command, refer to the section for that command.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Proxy.Action>
  <INPUT type="text" name="User.proxyId" size="50" value="John Doe">
  <INPUT Type="Checkbox" Name="Proxy.Access.Read.mail" Value="1">
  <INPUT type="image" name="Proxy.AddAccess"
    src="/com/novell/webaccess/images/btnsaveen.gif" alt="Save" |
    border=0>
</FORM>
```

Proxy.AddAccess

Adds a user to the proxy access list.

Parameters

Proxy.Access.Read.mail

Specifies to grant rights to read mail.

Proxy.Access.Write.mail

Specifies to grant rights to create mail

Proxy.Access.Read.note

Specifies to grant rights to read notes.

Proxy.Access.Wrtite.note

Specifies to grant rights to create notes.

Proxy.Access.Read.todo

Specifies to grant rights to read tasks.

Proxy.Access.Write.todo

Specifies to grant rights to create tasks.

Proxy.Access.Read.appointment

Specifies to grant rights to read appointments.

Proxy.Access.Write.appointment

Specifies to grant rights to create appointments.

Proxy.Access.Option.notify

Specifies to grant rights to notifications.

Proxy.Access.Read.private

Specifies to grant rights to read private items.

Proxy.Access.Option.settings

Specifies to grant rights to modify options, rules, and folders.

Proxy.Access.Option.alarms

Specifies to grant rights to alarms.

User.proxyId (Required)

Specifies the name of the user to add to the proxy access list.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Proxy.ListAccess](#) (page 290).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Proxy.Action>
  <INPUT type="text" name="User.proxyId" size="50" value="John Doe">
  <INPUT Type="Checkbox" Name="Proxy.Access.Read.mail" Value="1">
  <INPUT type="image" name="Proxy.AddAccess"
    src="/com/novell/webaccess/images/btnsaveen.gif" alt="Save"
    border=0>
</FORM>
```

Proxy.CloseListAccess

Closes the proxy access list.

Parameters

Proxy.CloseListAccess (Required)

Specifies to close the proxy access list.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

None.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action
&Proxy.CloseListAccess="> Close Proxy Access List </A>
```

Proxy.CloseListHistory

Closes the proxy history list.

Parameters

Proxy.CloseListHistory (Required)

Specifies to close the proxy history list.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

None.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action
&Proxy.CloseListHistory="> Close Proxy History List </A>
```

Proxy.DeleteAccess

Deletes a user from a proxy access list.

Parameters

Proxy.DeleteAccess (Required)

Specifies the unique identifier of the user in the proxy access list.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Proxy.ListAccess](#) (page 290).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action
&Proxy.DeleteAccess=1199"> Delete User in Access List </A>
```

Proxy.DeleteHistory

Deletes a user from a proxy history list.

Parameters

Proxy.DeleteHistory (Required)

Specifies the unique identifier of the user in the proxy history list.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Proxy.ListHistory](#) (page 291).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action
&Proxy.DeleteHistory=1213"> Delete User in History List </A>
```


Proxy.Directory

Integrates both the access and history proxy lists with the GroupWise Address Book. This action opens the GroupWise Address Book and must be specified as a subcommand of [Proxy.Action \(page 282\)](#).

Parameters

In addition to passing the parameters used by [Directory.List \(page 182\)](#), it might be helpful to pass the `Compose.id` into the process (template) doing the directory so that it will be available for adding recipients to the directory.

Compose.id

Specifies the identifier of the proxy item, as returned from `Proxy.Directory`.

Proxy.Directory (Required)

Specifies the required subaction.

User context (Required)

Specifies the context identifier for this user's session.

Return Values

If modifying the proxy access list, returns the same values as [Proxy.ListAccess \(page 290\)](#).

If modifying the proxy history list, returns the same values as [Proxy.ListHistory \(page 291\)](#).

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Proxy.Action>
  <INPUT type=hidden name=Compose.id value=1>
  <INPUT type=submit name=Proxy.Directory value="Address Book">
</FORM>
```

Proxy.ListAccess

Lists users in the proxy access list.

Parameters

Proxy.ListAccess (Required)

Specifies the list proxy access.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Proxy.Access.count	Number of users in the proxy access list.
Proxy.Access[n].id	Identifier of the user in the proxy access list.
Proxy.Access[n].name	Name of the <i>n</i> th user.
Proxy.Access[n].email	Email address of the <i>n</i> th user.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action  
&Proxy.ListAccess="> Proxy Access List </A>
```

Proxy.ListHistory

Lists users in the proxy history list.

Parameters

Proxy.ListHistory (Required)

Specifies the list proxy history.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Proxy.History.count	Number of users in the proxy history list.
Proxy.History[n].id	Identifier of the user in the proxy history list.
Proxy.History[n].name	Name of the <i>n</i> th user.
Proxy.History[n].email	Email address of the <i>n</i> th user.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action  
&Proxy.ListHistory="> Proxy History List </A>
```

Proxy.Login

Performs a proxy login to GroupWise.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

User.id (Required)

Specifies the identifier of the current user.

User.proxyId (Required)

Specifies the proxy to the user's account.

Return Values

Returns the same values as [User.Login \(page 331\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Login
&User.id=John Doe&User.proxyId=John Doe2"> Proxy Login </A>
```

Proxy.ModifyAccess

Modifies the rights of a user in a proxy access list.

Parameters

Proxy.Access.id (Required)

Specifies the identifier of the user being modified.

User.context (Required)

Specifies the context identifier for this user's session.

Proxy.Access.Read.mail

Specifies to grant rights to read mail.

Proxy.Access.Write.mail

Specifies to grant rights to create mail

Proxy.Access.Read.note

Specifies to grant rights to read notes.

Proxy.Access.Wrtite.note

Specifies to grant rights to create notes.

Proxy.Access.Read.todo

Specifies to grant rights to read tasks.

Proxy.Access.Write.todo

Specifies to grant rights to create tasks.

Proxy.Access.Read.appointment

Specifies to grant rights to read appointments.

Proxy.Access.Write.appointment

Specifies to grant rights to create appointments.

Proxy.Access.Option.notify

Specifies to grant rights to notifications.

Proxy.Access.Read.private

Specifies to grant rights to read private items.

Proxy.Access.Option.settings

Specifies to grant rights to modify options, rules, and folders.

Proxy.Access.Option.alarms

Specifies to grant rights to alarms.

Return Values

Returns the same values as [Proxy.ListAccess](#) (page 290).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Proxy.Action>
  <INPUT type="text" name="Proxy.Access.id" value="1216">
  <INPUT Type="Checkbox" Name="Proxy.Access.Read.mail" Value="1" >
  <INPUT type="image" name="Proxy.ModifyAccess"
    src="/com/novell/webaccess/images/btnsaveen.gif" alt="Save"
    border=0>
</FORM>
```

Proxy.ReadAccess

Reads the proxy access rights for a user.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Proxy.ReadAccess (Required)

Specifies the unique identifier of a user in the proxy access list.

Return Values

Value	Description
Proxy.Access.id	Identifier of the user in the proxy access list.
Proxy.Access.email	Email address in the proxy access list.
Proxy.Access.name	Name of the user in the proxy access list.
Proxy.Access.Read.mail	User has rights to read mail.
Proxy.Access.Write.mail	User has rights to create mail.
Proxy.Access.Read.note	User has rights to read notes.
Proxy.Access.Write.note	User has rights to create notes.
Proxy.Access.Read.todo	User has rights to read tasks.
Proxy.Access.Write.todo	User has rights to create tasks.
Proxy.Access.Read.appointment	User has rights to read appointments.
Proxy.Access.Write.appointment	User has rights to create appointments.
Proxy.Access.Option.notify	User has rights to notifications.
Proxy.Access.Read.private	User has rights to read private items.
Proxy.Access.Option.settings	User has rights to modify options, rules, and folders.
Proxy.Access.Option.alarms	User has rights to alarms.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Proxy.Action  
&Proxy.readAccess=1216"> Read Proxy Access </A>
```

10.12 Rules

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Directory.GetRecipients” on page 297](#)
- ◆ [“Rule.Action” on page 298](#)
- ◆ [“Rule.Cancel” on page 299](#)
- ◆ [“Rule.Create” on page 300](#)
- ◆ [“Rule.CreateType” on page 302](#)
- ◆ [“Rule.Delete” on page 303](#)
- ◆ [“Rule.Directory” on page 304](#)
- ◆ [“Rule.Enable” on page 306](#)
- ◆ [“Rule.List” on page 307](#)
- ◆ [“Rule.Modify” on page 308](#)
- ◆ [“Rule.Read” on page 310](#)

Directory.GetRecipients

Populates the shared folder list with a list of selected users retrieved from the GroupWise Address Book and closes the GroupWise Address Book. This action must be specified as a subcommand of [Folder.Action \(page 217\)](#).

Parameters

Directory.GetRecipients (Required)

Specifies the required subaction.

Compose.id (Required)

Specifies the identifier of the shared folder list in the GroupWise Address Book, as created with [Folder.Directory \(page 220\)](#).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Folder.ListShare \(page 227\)](#).

Remarks

Directory.GetRecipients is necessary only if you have used any of the following Directory Accumulator actions.

- ♦ [Directory.Accumulator.Add.BC \(page 158\)](#)
- ♦ [Directory.Accumulator.Add.CC \(page 159\)](#)
- ♦ [Directory.Accumulator.Add.To \(page 160\)](#)
- ♦ [Directory.Accumulator.Delete \(page 161\)](#)
- ♦ [Directory.Accumulator.List \(page 162\)](#)

Example

```
<FORM method=post action=/servlet/webacc>  
  <INPUT type=hidden name=User.context value=xyz>  
  <INPUT type=hidden name=action value=Folder.Action>  
  <INPUT type=hidden name=Compose.id value=1>  
  <INPUT type=image name=Directory.GetRecipients src="btokus.gif">  
</FORM>
```

Rule.Action

Performs a specified action.

Parameters

One of the following Rule commands must be specified in the URL or appear as the name of a form object:

[Directory.GetRecipients \(page 215\)](#)

[Rule.Cancel \(page 299\)](#)

[Rule.Create \(page 300\)](#)

[Rule.CreateType \(page 302\)](#)

[Rule.Delete \(page 303\)](#)

[Rule.Directory \(page 304\)](#)

[Rule.Enable \(page 306\)](#)

[Rule.List \(page 307\)](#)

[Rule.Modify \(page 308\)](#)

[Rule.Read \(page 310\)](#)

The existence of one of these values in the URL determines the action that is invoked. For information about the parameters and values of a command, refer to the section for that command.

Return Values

Returns the result of the rule command used as a parameter. For information about the return value of a command, refer to the section for that command.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Rule.Action
&Rule.Read=&Rule.drn=1"> Read Rule </A>
```

Rule.Cancel

Cancels the creation or modification of a rule. This action must be specified as a subcommand of [Rule.Action \(page 298\)](#).

Parameters

Compose.id (Required)

Specifies the identifier of the rule item, as returned from [Rule.CreateType \(page 302\)](#) or [Rule.Read \(page 310\)](#).

Rule.Cancel (Required)

Specifies the required subcommand.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Rule.List \(page 307\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Rule.Action>
  <INPUT type=hidden name= Rule.Cancel value=1 >
  <INPUT type=hidden name=Compose.id value=1>
    src="btncancelus.gif">
</FORM>
```

Rule.Create

Saves a rule.

Parameters

The following parameters are for all rules. A rule parameter might be required only if you are saving a specific rule type. For example, Folder.id or Folder.drn is required if you are saving a Move to Folder type of rule.

User.context (Required)

Specifies the context identifier for this user's session.

Compose.id (Required)

Specifies the identifier of the rule item, as returned from [Rule.CreateType \(page 302\)](#) or [Rule.Read \(page 310\)](#).

Folder.id or Folder.drn

Specifies the folder where items are moved.

Item.bc

Specifies the list of BC recipients.

Item.cc

Specifies the list of CC recipients.

Item.message

Specifies the message text.

Item.subject

Specifies the message subject.

Item.to

Specifies the list of TO recipients.

RuleConditioncondition

Specifies the type of condition: Contains, Does Not Contain, Begins With, Matches.

RuleConditionfield

Specifies the GroupWise field that is used for rule comparison: From, Message, Subject, To.

Rule.Condition.text

Specifies the comparison text for the rule condition.

Rule.drn

Specifies the unique identifier of a rule.

Rule.name (Required)

Specifies the rule name.

Rule.replyTo

Specifies the Reply To mode: 0 Reply to Sender and 1 Reply to All.

Rule.subjectPrefix

Specifies the prefix for the subject field: Fwd:, Re:.

Rule.type (Required)

Specifies the rule type: Delete, Forward, Move to Folder, Reply, and Vacation.

Return Values

Returns the same values as [Rule.List \(page 307\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Rule.Action>
  <INPUT type=hidden name= Rule.Create value=1 >
  <INPUT type=hidden name=Compose.id value=2>
  <INPUT type=hidden name=Rule.type value="vacation">
  <INPUT type=text name=Rule.name value="Vacation Rule">
  <INPUT type=text name=Item.subject value="Out of Office">
  <TEXTAREA Name="Item.message" rows=1 cols=1 ></TEXTAREA>
    src="btnsaveus.gif">
</FORM>
```

Rule.CreateType

Opens a rule template of the specified type.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Rule.Type (Required)

Specifies the rule type:

Delete

Forward

Move to Folder

Reply

Vacation

Return Values

Returns a rule template based on the selected rule type.

Example

```
<FORM method=post action=/servlet/webacc>  
  <INPUT type=hidden name=User.context value=xyz>  
  <INPUT type=hidden name=action value=Rule.Action>  
  <INPUT type=hidden name=Rule.Type value="Vacation">  
    src="btncreateus.gif">  
</FORM>
```

Rule.Delete

Deletes the selected rule.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Rule.Delete (Required)

Specifies the required action.

Rule.drn (Required)

Specifies the unique rule identifier.

Return Values

Returns the same values as [Rule.List \(page 307\)](#).

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Rule.Action
&Rule.Delete=&Rule.drn=1"> Delete Rule </A>
```

Rule.Directory

Populates the TO, CC, and BC fields in rules from the GroupWise Address Book. This action opens the GroupWise Address Book and must be specified as a subcommand of the [Rule.Action \(page 298\)](#) action.

Parameters

In addition to the parameters used by [Directory.List \(page 182\)](#), it might be helpful to pass the Compose.id into the process (template) doing the directory so that it will be available for adding recipients to the directory.

Rule.Directory (Required)

Specifies the required action.

Compose.id (Required)

Specifies the identifier of the rule item, as returned from [Rule.CreateType \(page 302\)](#) or [Rule.Read \(page 310\)](#).

User context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns to the rule template. Information that has been added to the rule fields is returned.

Value	Description
Folder.id	Move items to this folder.
Folder.drn	Move items to this folder.
Item.bc	List of BC recipients.
Item.cc	List of CC recipients.
Item.message	Message text.
Item.subject	Message subject.
Item.to	List of TO recipients.
RuleCondition	Type of condition: Contains, Does Not Contain, Begins With, Matches.
RuleConditionfield	GroupWise field used for rule comparison: From, Message, Subject, To.
Rule.Condition.text	Comparison text for the rule condition.
Rule.drn	Unique identifier of the rule.
Rule.name	Rule name.
Rule.replyTo	0 = Reply to Sender, 1 = Reply to All.
Rule.subjectPrefix	Prefix for the subject field: Fwd:, Re:.
Rule.type	Rule type: Delete, Forward, Move to Folder, Reply, Vacation.

Example

```
<FORM method=post action="/servlet/webacc">  
  <INPUT type=hidden name=User.context value=xyz>  
  <INPUT type=hidden name=action value=Rule.Action>  
  <INPUT type=hidden name=Compose.id value=1>  
  <INPUT type=submit name=Rule.Directory value="Address Book">  
</FORM>
```

Rule.Enable

Enables or disables existing rules. Rules are enabled (activated) by passing the rule DRN with the Rule.Enable command.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Rule.Enable (Required)

Specifies the required action.

Rule.drn (Required)

Specifies the unique rule identifier (for example, 113, 115, etc.).

Return Values

Returns the same values as [Rule.List \(page 307\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Rule.Action>
  <INPUT type=hidden name=Rule.Enable value=1>
  <INPUT type=checkbox name=Rule.drn value=113>
    src="btnsaveus.gif">
</FORM>
```

Rule.List

Lists the rules.

Parameters

Rule.List (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Rule.count	Number of rules.
Rule[n].drm	Unique identifier of the rule.
Rule[n].name	Name of the rule.
Rule[n].editable	If true, the rule can be edited in WebAccess.
Rule[n].enabled	If true, the rule is enabled.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Rule.Action&Rule.List="> List  
Rules  
</A>
```

Rule.Modify

Modifies an existing rule.

Parameters

The following parameters are for all rules. A rule parameter might be required only if you are saving a specific rule type. For example, Folder.id or Folder.drn is required if you are saving a "Move to Folder" type of rule.

User.context (Required)

Specifies the context identifier for this user's session.

Compose.id

Specifies the identifier of the rule item as returned from Rule.Modify.

Folder.id or Folder.drn

Specifies the folder where items are moved.

Item.bc

Specifies the list of BC recipients.

Item.cc

Specifies the list of CC recipients.

Item.message

Specifies the message text.

Item.subject

Specifies the message subject.

Item.to

Specifies the list of To recipients.

RuleCondition

Specifies the type of condition: Contains, Does Not Contain, Begins With, Matches.

RuleConditionfield

Specifies the GroupWise field used for rule comparison: From, Message, Subject, To.

Rule.Condition.text

Specifies the comparison text for the rule condition.

Rule.drn

Specifies the unique identifier of a rule.

Rule.name (Required)

Specifies the rule name.

Rule.replyTo

Specifies the Reply To mode: 0 (reply to sender) and 1 (reply to all).

Rule.subjectPrefix

Specifies the prefix for the subject field (for example: Fwd:, Re:).

Rule.type (Required)

Specifies the rule type: Delete, Forward, Move to Folder, Reply, and Vacation.

Return Values

Returns the same values as [Rule.List \(page 307\)](#).

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Rule.Action>
  <INPUT type=hidden name= Rule.Modify value=1 >
  <INPUT type=hidden name=Compose.id value=2>
  <INPUT type=hidden name=Rule.type value="vacation">
  <INPUT type=text name=Rule.name value="Vacation Rule">
  <INPUT type=text name=Item.subject value="Out of Office">
  <TEXTAREA Name="Item.message" rows=1 cols=1 ></TEXTAREA>
    src="btnsaveus.gif">
</FORM>
```

Rule.Read

Returns the details of a selected rule.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Rule.Read (Required)

Specifies the required action.

Rule.drn (Required)

Specifies the unique rule identifier.

Return Values

Value	Description
Folder.id	Move items to this folder.
Folder.drn	Move items to this folder.
Item.bc	List of BC recipients.
Item.cc	List of CC recipients.
Item.message	Message text.
Item.subject	Message subject.
Item.to	List of To recipients.
RuleCondition	Type of condition (for example, Contains, Does Not Contain, Begins With, Matches).
RuleConditionfield	GroupWise field used for rule comparison (for example, From, Message, Subject, To).
Rule.Condition.text	Comparison text for the rule condition.
Rule.drn	Unique identifier of the rule.
Rule.name	Rule name.
Rule.replyTo	Reply To mode. Possible values are 0 (reply to sender) and 1 (reply to all).
Rule.subjectPrefix	Prefix for the subject field (for example, Fwd:, Re:).
Rule.type	Rule type. Possible values include: Delete, Forward, Move to Folder, Reply, and Vacation.

Example

```
<A href="/servlet/webacc?User.context=xyz&action= Rule.Action
&Rule.Read=&Rule.drn=1"> Read Rule </A>
```

10.13 Security

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Security.ChangePassword” on page 312](#)

Security.ChangePassword

Changes the user's password.

Parameters

User.oldPassword (Required)

Specifies the old password.

User.newPassword (Required)

Specifies the new password.

User.confirmPassword (Required)

Specifies the new password for confirmation (must match User.newPassword or the password is not changed).

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

None.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type=hidden name=User.context value=xyz>
  <INPUT type=hidden name=action value=Security.ChangePassword>
  Old Password:
  <INPUT type=password name=User.oldPassword value=><BR>
  New Password:
  <INPUT type=password name=User.newPassword value=""><BR>
  Confirm Password:
  <INPUT type=password name=User.comfirmPassword value=""><BR>
</FORM>
```


10.14 Send Options

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“SendOptions.Modify” on page 314](#)
- ◆ [“SendOptions.Read” on page 317](#)

SendOptions.Modify

Modifies the global send options for the user.

Parameters

SendOptions.Modify (Required)

Specifies the required action.

SendOptions.priority

Specifies the priority of the item: Low, Medium, High.

SendOptions.security

Specifies the security settings on the item:

- 1 Normal
- 2 Proprietary
- 3 Confidential
- 4 Secret
- 5 Top Secret
- 6 For your eyes only

SendOptions.ReplyRequested.day

Specifies the number of days selected for a reply request.

SendOptions.ReplyRequested.type

Specifies the type of reply requested: WhenConvenient, ByDate.

SendOptions.ReturnNotification.accepted

Specifies the type of return notification for an accepted item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

SendOptions.ReturnNotification.completed

Specifies the type of return notification for a completed item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

SendOptions.ReturnNotification.deleted

Specifies the type of return notification for a deleted item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

SendOptions.ReturnNotification.declined

Specifies the type of return notification for a declined item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

SendOptions.ReturnNotification.opened

Specifies the type of return notification for an opened item:

- 1 None
- 2 Mail
- 4 Notify
- 8 Notify and Mail

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
SendOptions.priority	Item priority: Low, Medium, High.
SendOptions.security	Security settings on the item: <ul style="list-style-type: none">1 Normal2 Proprietary3 Confidential4 Secret5 Top Secret6 For your eyes only
SendOptions.ReplyRequested.day	Number of days selected for a reply request
SendOptions.ReplyRequest.type	Type of reply requested: WhenConvenient, ByDate
SendOptions.ReturnNotification.accepted	Type of return notification for an accepted item: <ul style="list-style-type: none">1 None2 Mail4 Notify8 Notify and Mail
SendOptions.ReturnNotification.completed	Type of return notification for a completed item. <ul style="list-style-type: none">1 None2 Mail4 Notify8 Notify and Mail

Value	Description
SendOptions.ReturnNotification.deleted	Type of return notification for a deleted item. 1 None 2 Mail 4 Notify 8 Notify and Mail
SendOptions.ReturnNotification.declined	Type of return notification for a declined item. 1 None 2 Mail 4 Notify 8 Notify and Mail
SendOptions.ReturnNotification.opened	Type of return notification for an opened item. 1 None 2 Mail 4 Notify 8 Notify and Mail
SendOptions.saveMsg	Flag equals 1 if the send options saved without errors.

Example

```
<FORM method=post action="/servlet/webacc">
  <INPUT type=hidden name=User.context value=xyz>
  .
  .
  <INPUT type=submit name=SendOptions.Security value="High">
  .
  .
  <INPUT type=submit name=SendOptions.Modify value="1">
</FORM>
```

SendOptions.Read

Reads the global send options for the user.

Parameters

SendOptions.Read (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
SendOptions.priority	Item priority: Low, Medium, High.
SendOptions.security	Security settings on the item: 1 Normal 2 Proprietary 3 Confidential 4 Secret 5 Top Secret 6 For your eyes only
SendOptions.ReplyRequested.day	Number of days selected for a reply request
SendOptions.ReplyRequest.type	Type of reply requested: WhenConvenient, ByDate
SendOptions.ReturnNotification.accepted	Type of return notification for an accepted item: 1 None 2 Mail 4 Notify 8 Notify and Mail
SendOptions.ReturnNotification.completed	Type of return notification for a completed item. 1 None 2 Mail 4 Notify 8 Notify and Mail
SendOptions.ReturnNotification.deleted	Type of return notification for a deleted item. 1 None 2 Mail 4 Notify 8 Notify and Mail

Value	Description
SendOptions.ReturnNotification.declined	Type of return notification for a declined item. 1 None 2 Mail 4 Notify 8 Notify and Mail
SendOptions.ReturnNotification.opened	Type of return notification for an opened item. 1 None 2 Mail 4 Notify 8 Notify and Mail

Example

```
<FORM method=post action="/servlet/webacc">  
  <INPUT type=hidden name=User.context value=xyz>  
  <INPUT type=submit name=SendOptions.Read value="1">  
</FORM>
```

10.15 Settings

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Settings.General.Modify” on page 320](#)
- ◆ [“Settings.Get” on page 321](#)
- ◆ [“Settings.Set” on page 322](#)

Settings.General.Modify

Sets general WebAccess settings for a user. For example, `Settings.MessageList.count` stores the number of messages a user wants displayed in the message list.

Parameters

Setting.General.Modify (Required)

Specifies the required action.

Setting.MessageList.count (Required)

Specifies the number of messages a user wants to display in the message list.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Settings.Get \(page 321\)](#).

Returns `Settings.MessageList.count` with a value.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Settings.General.Modify&merge=general&Settings.MessageList.count=20">Message List Count</A>
```


Settings.Get

Retrieves a WebAccess-only setting generically.

Parameters

Setting.name (Required)

Specifies the name of the setting.

Setting.Get (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Setting[n].name	Name of the <i>n</i> th setting.
Setting[n].value	Value of the <i>n</i> th setting.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Setting.Get&merge=general&Setting.name=General.Setting.ItemCount"> Item Count</A>
```

Settings.Set

Sets a WebAccess-only setting generically.

Parameters

Setting.name (Required)

Specifies the name of the setting.

Setting.Set (Required)

Specifies the required action.

Setting.value (Required)

Specifies the value of the setting.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [Settings.Get \(page 321\)](#).

Remarks

To delete an existing setting, pass in Setting.name without a value.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Setting.Set&merge=general&Setting.name=General.Setting.ItemCount&Setting.value=25">Item Count</A>
```

10.16 Signature

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“Signature.Modify” on page 324](#)
- ◆ [“Signature.Read” on page 325](#)

Signature.Modify

Modifies a user's signature.

Parameters

Signature.isEnabled

Specifies the signature is enabled if this parameter is included. Otherwise, the signature is disabled.

Signature.isAutomatic (Required)

Specifies if the signature is automatic:

automatic
prompt

Signature.signature (Required)

Specifies the signature text.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Signature.isEnabled	A signature is enabled or disabled.
Signature.isAutomatic	A signature can be automatically added or the user can be prompted to add a signature to outgoing messages.
Signature.saveMsg	If the signature saved without errors, the flag equals 1.
Signature.signature	Signature text.

Example

```
<FORM method=post action=/servlet/webacc>  
  <INPUT type=hidden name=User.context value=xyz>  
  <INPUT type=hidden name=action value=Signature.Modify>  
  <INPUT type=checkbox name="Signature.isEnabled" value="enabled">  
  <TEXTAREA Name="Signature.signature" Rows="4" Cols="40"></TEXTAREA>  
  <INPUT type=radio name="Signature.isAutomatic" value="automatic">  
  <INPUT type=radio name="Signature.isAutomatic" value="prompt">  
</FORM>
```

Signature.Read

Reads a user's signature.

Parameters

Signature.Read (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
Signature.isEnabled	A signature is enabled or disabled.
Signature.isAutomatic	A signature can be automatically added or the user can be prompted to add a signature to outgoing messages.
Signature.signature	Signature text.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Signature.Read&merge=signature"> Read Signature </A>
```

10.17 Time Zone

Click any action name to view the description, parameters, and return values for that action.

- ♦ [“TimeZone.List” on page 327](#)
- ♦ [“TimeZone.Modify” on page 328](#)
- ♦ [“TimeZone.Read” on page 329](#)

TimeZone.List

Writes out a list of time zones supported in GroupWise. It also writes out selected time zone information.

Parameters

TimeZone.List (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
TimeZone.id	Selected time zone ID.
TimeZone[n].name	TimeZone name of the <i>n</i> th item.
TimeZone[n].tableId	TimeZone table ID of the <i>n</i> th item.
User.TimeZone.offset	Selected time zone offset from GMT.
User.TimeZone.dst	True if the user is in daylight saving time. Otherwise, false.
User.TimeZone.desc	Standard or daylight description for the current time zone.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=
TimeZone.List&merge=timezone"> List TimeZones </A>
```

TimeZone.Modify

Modifies the current time zone to the selected time zone.

Parameters

TimeZone.Modify (Required)

Specifies the required action.

TimeZone.id (Required)

Specifies the ID of the time zone: 1-74.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Returns the same values as [TimeZone.List \(page 327\)](#).

Example

```
<FORM action="{VAR Home.url}" method=post">
  <INPUT type="hidden" name="User.context" value="{VAR User.context}">
  <INPUT type="hidden" name="action" value="TimeZone.Modify">
  <INPUT type="hidden" name="merge" value="timezone">
  <INPUT type="hidden" name="error" value="timezone">
  <SELECT Name="TimeZone.id">
    <OPTION value="{VAR TimeZone[a].tableId}" {IF TimeZone.id ==
      a}SELECTED{ENDIF}>{VAR TimeZone[a].name}</OPTION>
  </SELECT>
</FORM>
```


TimeZone.Read

Reads the user's time zone information.

Parameters

TimeZone.Read (Required)

Specifies the required action.

User.context (Required)

Specifies the context identifier for this user's session.

Return Values

Value	Description
TimeZone.id	Selected time zone ID.
User.TimeZone.offset	Selected time zone offset from GMT.
User.TimeZone.dst	True if the user is in daylight saving time. Otherwise, false.
User.TimeZone.desc	Standard or daylight description for the current time zone.

Remarks

If the user has not set a time zone, the time zone information for the post office is used. The user's time zone is initialized during login.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=
TimeZone.Read&merge=timezone">Read TimeZone </A>
```

10.18 User Authentication

Click any action name to view the description, parameters, and return values for that action.

- ◆ [“User.Login” on page 331](#)
- ◆ [“User.Logout” on page 333](#)

User.Login

Logs in to GroupWise.

Parameters

User.id (Required)

Specifies the user ID to use during authentication.

User.password (Required)

Specifies the user password to use during authentication.

GWAP.ip

Specifies the overriding IP address of GroupWise WebAccess Agent.

GWAP.port

Specifies the overriding port of GroupWise WebAccess Agent.

GWAP.version

Specifies the version of GroupWise WebAccess Agent: 8.0, 8.0.1, 8.0.2, 8.0.3.

User.interface

Specifies the name of the directory (a subdirectory beneath the Templates.path directory that is specified in the configuration file) containing the templates that generate the user interface for this session.

User.lang

Specifies the two-letter ISO code of the desired language for the user interface.

User.context

Specifies the context identifier for this user's session. If specified, WebAccess attempts to use this context as the identifier for the user. If not specified, a new context is assigned.

User.displayDraftItems

Specifies that the interface supports draft items.

NOTE: Although User.displayDraftItems can be set in the templates at log in, the provider and GroupWise WebAccess agent (GWINTER) must also support the feature. If supported, User.displayDraftItems is set to 1 and made available as a User/Global variable by the provider. After logging in, this flag should be used to enable or disable draft item related UI items (for example, display the Work in Progress folder or the draft items in message lists, etc.).

Return Values

Value	Description
User.id	User login ID (for example, jsmith).
User.name	User's name (for example, John Smith).
User.fullId	Fully qualified name of the user (for example, jsmith.p01.domain1).

Value	Description
User.context	Context identifier assigned to this user's session.
User.Proxy.id	User login ID of the user account that is being proxied.
Login.grace	1 = Logging in used a grace login. The password should be changed soon.
User.Disk.limit	Amount of disk space allowed for this user's account.
User.Disk.threshold	Threshold at which to warn the user that the disk space limit has almost been reached.
User.LDAP.authenticated	1 = LDAP authentication used.
GWAP.Disk.inUse	Amount of disk space in use by this user's account.
Timezone.id	Specifies the ID of the time zone.
User.TimeZone.offset	
User.TimeZone.dst	1 = Time zone is currently in daylight savings. 0 = Time zone is not in daylight savings.
User.TimeZone.desc	String describing the time zone (for example, Eastern Standard Time).

Example

```
<A href="http://server:port/servlet/webacc?action=User.Login&User.context=xyz&User.name=jsmith&User.password=abc5def">Login</A>
```

User.Logout

Logs out of GroupWise.

Parameters

User.context (Required)

Specifies the context identifier for this user's session.

Example

```
<A href=" /servlet/webacc?action=User.Logout&User.context=xyz"> Logout </A>
```

11 LDAP Provider Tasks

WebAccess LDAP Provider is a component of GroupWise WebAccess that uses Light-weight Directory Access Protocol (LDAP) to retrieve information and perform operations.

This document describes the commands and parameters that can be used in URLs and HTML forms to allow access to LDAP GroupWise from a Browser in the following sections:

- ♦ [Section 11.1, “Calling the LDAP Provider,” on page 335](#)
- ♦ [Section 11.2, “Calling the Servlet,” on page 335](#)
- ♦ [Section 11.3, “Adding Additional LDAP Directories,” on page 335](#)
- ♦ [Section 11.4, “Specifying a Template,” on page 336](#)

11.1 Calling the LDAP Provider

- To call the provider, include in the URL or a [hidden] form field, the word “action” accompanied by the name of the to action perform, such as Directory.Search. (Also include the name of the necessary parameters and their values.)

The name of the LDAP provider is LDAP. This name must be provided as the value for the Provider.name parameter so that the servlet will call the LDAP provider.

11.2 Calling the Servlet

Actions and their associated parameters are processed by the WebAccess servlet, which is installed when you install GroupWise WebAccess.

- To call the servlet, create a URL that references the IP address or Host name of the computer where the servlet was installed (the workstation that is running your HTTP server) and specify /servlet/webacc.

For example, `http://151.123.456.78/servlet/webacc`.

11.3 Adding Additional LDAP Directories

To add LDAP directories, include the necessary information in the ldap.cfg LDAP configuration file, as in the following sample:

```

#####$#####
#Novell WebAccess LDAP Provider Configuration File
#####
#####
# sample configuration
# the ".n" at the end of each field represents that all of the fields
# describe the "nth" provider as listed in this file.
#####Provider.LDA
P.Config.name.n = <Name of the LDAP service>
Provider.LDAP.Config.server.n = <IP Address or host name of the server
                                providing the LDAP service>
Provider.LDAP.Config.port.n = <Port of the server providing the LDAP
                                service (usually 389)>
Provider.LDAP.Config.maxResults.n = <Maximum number of results to be
                                returned.>
Provider.LDAP.Config.timeout.n = <Number of seconds before timing
                                out>
Provider.LDAP.Config.user.n = <User id used to authenticate to
                                the LDAP server>
Provider.LDAP.Config.password.n = <User password
                                used to authenticate to LDAP server>
Provider.LDAP.Config.searchBase.n = <Hierarchical base where search
                                begins>
Provider.LDAP.Config.map.n = <Comma delimited list of name value
                                pairs in the form of name=value.
                                The name portion represents the
                                Keyword used by the LDAP server.
                                The value portion represents the
                                name that appears in the template.>
Provider.LDAP.Image.url.n = <URL of the image or logo for this
                                LDAP service>
Provider.LDAP.Web.url.n = <URL of the LDAP service
                                provider's Web site>
#####

```

11.4 Specifying a Template

When a URL is clicked or form is submitted, the browser sends the action to the HTTP server which in turn calls the WebAccess servlet. If action is specified, the servlet call the providers, the generates the resulting HTML page using a template. If no template is specified, Login.htm is used.

- To specify a template, include "merge" accompanied by the name of the template to be used.

Templates always end with an extension of .htm, but the extension is not included when specifying the name of the template. For example, "...&merge=addrbook" will use the template addrbook.htm to generate the HTML that will be sent back to the browser.

```
<A href="/servlet/webacc?Provider.name=LDAP&action=Directory.Search
&merge=addrBook">Login</A>
```

```
<FORM method=post action=/servlet/webacc>
<INPUT type=hidden name="Provider.name" value="GWAP">
<INPUT type=hidden name="merge" value="webacc">
<INPUT type=hidden name="action" value="User.Login">
User Id:
<INPUT type=input name="User.id" value=""><BR>
Password:
<INPUT type=password name="User.password" value=""><BR>
</FORM>
```

Provider.name is not necessary since GWAP is the default provider for the WebAccess servlet.

12 LDAP Provider Methods

The following directory methods operate on GroupWise directories (AddressBooks):

- ♦ [“Directory.Action” on page 338](#)
- ♦ [“Directory.List” on page 339](#)
- ♦ [“Directory.Search” on page 340](#)

Directory.Action

Indicates the presence of a parameter whose name is one of the Directory commands, and whose value may or may not be present.

Parameters

One of the following Directory commands must be specified in the URL or appear as the name of a form object:

[Directory.List \(page 339\)](#)

[Directory.Search \(page 340\)](#)

Refer to the section for each of these commands for the parameters that each they use.

Returns

Refer to the section for each of these commands for the results that they each return.

Example

```
<FORM method="post" action=/servlet/webacc>
  <INPUT type="hidden" name="Directory.Action">
  <INPUT type="submit" name="Directory.Modify">
  <INPUT type="submit" name="Directory.Delete">
  . . .
</FORM>
```

Directory.List

Lists the available LDAP directories.

Parameters

Directory.queryCount

Specifies the maximum number of directories to retrieve.

User.context (Required)

Specifies the context identifier for this user's session.

Returns

The following table lists return values along with descriptions.

Directory.count	Number of directories retrieved.
Directory.id	Identifier for the directory that was searched.
Directory.name	Name of the directory that was searched.
Directory.isEmpty	Specifies if the directory is empty.
Directory.isReadOnly	Specifies if you have only read access to this directory.

Example

```
<A href="/servlet/webacc?User.context=xyz&action=Directory.List&Directory.queryCount=25"> Delete </A>
```

Directory.Search

Searches the specified directory for the specified entries.

Parameters

Directory.id (Required)

Specifies the identifier of the directory to search.

Directory.queryCount

Specifies how many directory items to retrieve.

Directory.queryType

Specifies whether to query from the beginning of the list or the current position:

AddressBook.queryFirstDEFAULT

AddressBook.queryNext

Directory.queryFirst

Specifies the first name to use for the query.

Directory.queryLast

Specifies the last name to use for the query.

User.context (Required)

Specifies the context identifier for this user's session.

Returns

The following table lists return values along with descriptions.

Directory.hasMore	There are more matches than were returned.
Directory.id	Identifier for the directory that was searched.
Directory.Item.count	Number of items returned from the search.
Directory.Item[n].id	Identifier of the <i>n</i> th item.
Directory.Item[n].email	Email address of the <i>n</i> th item.
Directory.Item[n].emailDisplay	Displayable version of the email address of the <i>n</i> th item.
Directory.Item[n].type	Item type of the <i>n</i> th item.
Directory.Item[n].name	Full name of the <i>n</i> th item.
Directory.Item[n].firstName	First name of the <i>n</i> th item.
Directory.Item[n].lastName	Last name of the <i>n</i> th item.
Directory.Item[n].phone	Telephone number of the <i>n</i> th item.
Directory.Item[n].fax	Fax number of the <i>n</i> th item.
Directory.Item[n].department	Department of the <i>n</i> th item.
Directory.Item[n].title	Title of the <i>n</i> th item.

Directory.name	Name of the directory that was searched.
Directory.isEmpty	Specifies if the directory is empty.
Directory.isReadOnly	Specifies if you have only read access to this directory.

Example

```
<FORM method=post action=/servlet/webacc>
  <INPUT type="hidden" name="User.context" value="xyz">
  <INPUT type="hidden" name="action" value="Directory.Search">
  <INPUT type="hidden" name="Directory.id"
    value="Frequent Contacts">
  <INPUT type="input" name="Directory.queryFirst" value="">
  <INPUT type="input" name="Directory.queryLast" value="">
  <INPUT type="submit" name="submit" value="Search">
</FORM>
```

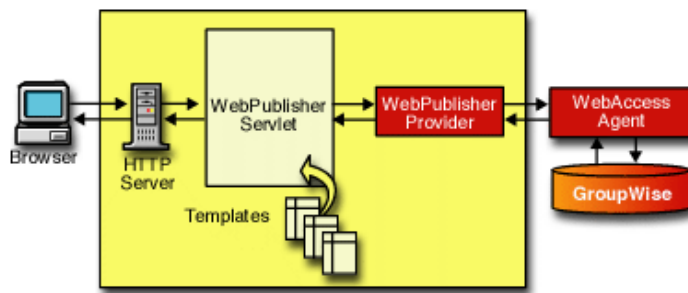

13 WebPublisher Provider Tasks

Novell WebPublisher Provider interfaces with GroupWise to retrieve information and perform operations.

This document describes various tasks that are necessary to allow a browser access to documents available to WebPublisher in the following sections:

- ♦ [Section 13.1, “Calling the WebPublisher Provider,”](#) on page 343
- ♦ [Section 13.2, “Communicating With the Servlet,”](#) on page 344
- ♦ [Section 13.3, “Specifying a Template,”](#) on page 344

Figure 13-1 WebPublisher Architecture



The architecture of WebPublisher is as follows:

The WebAccess agent communicates with the WebPublisher Provider, which uses the WebPublisher servlet in conjunction with templates to provide content to a browser over HTTP.

13.1 Calling the WebPublisher Provider

- ❑ To call the provider, include in the URL (or in a [hidden] form field), the word “action” accompanied by the name of the to action perform (such as Library.List).

Also include in the URL or other form fields, the name of the necessary parameters and their values.

The name of the WebPublisher provider is GWDoc. When WebAccess is installed, GWDoc is specified as the default provider. Since it is the default provider, it is not necessary to specify the name of the provider in the URL or forms. If the default provider is changed (in `webpub.cfg`), you must include the `Provider.name` parameter with all requests so that the servlet will call the WebPublisher provider.

13.2 Communicating With the Servlet

Actions and their associated parameters are processed by the WebPublisher servlet, which is installed when you install GroupWise WebPublisher.

- ❑ To call the servlet, create a URL that references the IP address or host name of the computer where the servlet was installed (this is the workstation that is running your HTTP server) and specify `/servlet/webpub`.

For example: `http://151.123.456.78/servlet/webpub`

13.3 Specifying a Template

When a URL is clicked or a form is submitted, the browser sends the action to an HTTP server, which in turn calls the WebPublisher servlet. If action is specified, the servlet calls the provider and then uses a template to generate the resulting HTML page. If no template is specified, `start.htm` is used.

- ❑ To specify a template, include the word “merge” accompanied by the name of the template to be used.

Templates always end with an `.htm` extension, but the extension is not included when specifying the name of the template. For example, “. . .&merge=liblist” uses the `liblist.htm` template file to generate the HTML that will be sent back to the browser.

See the Template Documentation and the following two examples for more information on templates.

```
<A href="/servlet/webpub?Provider.name=GWDoc&action=Library.List&merge=liblist">Login</A>
```

```
<FORM method=post action=/servlet/webpub>
  <INPUT type=hidden name="Provider.name" value="GWDoc">
  <INPUT type=hidden name="merge" value="liblist">
  <INPUT type=hidden name="action" value="Library.List">
  <INPUT type=submit name="submit" value="Submit">
</FORM>
```

`Provider.name` is not necessary in the previous example since `GWDoc` is the default provider for the `WebAccess` servlet.

14 WebPublisher Provider Methods

This section describes the following methods that can be used in URLs and HTML forms to allow access to documents available to WebPublisher from a browser:

- ♦ [“Document.Profile” on page 346](#)
- ♦ [“Document.Save” on page 347](#)
- ♦ [“Document.View” on page 348](#)
- ♦ [“Library.List” on page 349](#)
- ♦ [“Library.Search” on page 350](#)

Document.Profile

Retrieves the profile information for the specified document.

Parameters

Library.id(Required)

Specifies the library identifier of the desired document.

Document.id(Required)

Specifies the document number of the desired document.

Document.version (Required)

Specifies the desired numeric version number of the document, or one of the following values:

Official
Current

Returns

Library.id	Library identifier for this document.
Document.id	Document number for this document.
Document.version	Document version.
Document.author	Author of the document.
Document.extension	Document file extension (such as doc).
Document.size	Size of the document.
Document.subject	Document subject.
Document.Profile.createDate	Creation date of the document.
Document.Profile.modifyDate	Modification date of the document.
Document.Profile.officialVersion	Version number of the official version of the document.
Document.Profile.currentVersion	Version number of the current version of the document.
Document.Profile.versionCount	Number of version for this document.

Example

```
<A href="/servlet/webpub?User.context=xyz&action=
Document.Profile&Library.id=Library1&Document.id=
3452&Document.version=Official">Document Properties</A>
```

Document.Save

Downloads the specified document for saving.

Parameters

Provider.name

Specifies GWDOC for the name of the GroupWise WebPublisher provider.

Item.type

Specifies DREF for the Document Reference type.

Library.id

Specifies the identifier of the document library.

Document.id

Specifies the document number.

Document.version

Specifies the document number, or one of the following values:

Official

Current

Document.format

Specifies the GWNative format.

Document.section

Specifies the GWContentRoot section.

Returns

Returns the original contents of the document.

Remarks

Saving a document from a document reference requires the parameters to be specified in the “extra path info” of the URL. This is the portion of the URL that follows the name of the servlet and occurs before the question mark.

The parameters need to occur in a specific order as follows:

```
/servlet/webacc/User.context/Provider/Item.type/Library.Id/Document.id/  
Download.version/Document.format/Document.section
```

Example

```
<A href="/servlet/webpub/xyz/GWDOC/DREF/Lib1/6542/Official/GWNative/  
GWContentRoot?action=Document.Save"> Save </A>
```

Document.View

Downloads a document for viewing.

Returns

Returns an HTML version of the document if a conversion is possible.

Remarks

Document.View is invoked the same way as Document.Save except for the specific action name.

See Also

[Document.Save \(page 347\)](#)

Example

```
<A href="/servlet/webpub/xyz/GWDOC/DREF/Lib1/6542/Official/GWNative/  
GWContentRoot?action=Document.View"> View Document </A>
```

Library.List

Lists the available document libraries.

Parameters

Library.queryCount

Specifies the number of libraries to return.

Returns

Library.count	Number of document libraries.
Library.Default.id	Identifier of the user's default library.
Library[n].id	Identifier for the <i>n</i> th document library.
Library[n].name	Name of the <i>n</i> th document library.
Library[n].description	Description of the <i>n</i> th document library.

Example

```
/servlet/webpub?User.context=xyz&action=Library.List
```

Library.Search

Searches the specified document library.

Parameters

Library.queryField

Specifies which fields to search:

ALLSearch all fields (Default)

AUTHORSearch author fields

SUBJECTSearch subject fields

Library.queryId

Specifies the name of the library to search (can repeat to search as many libraries as desired).

Library.queryCount

Specifies how many documents to retrieve.

Library.queryType:

Specifies whether to query from the beginning of the list or the current position (defaults to the beginning of the list):

Document.queryFirst Start from beginning

Document.queryNext Start from the current position

Library.queryText:

Specifies the text to use for the query. If not present (or blank), all documents will be found.

Returns

Library.queryCount	Number of document requested.
Library.queryField	Search field used.
Library.queryText	Search criteria.
Library.queryId	Libraries searched.
Library.hasMore	1 = More documents were found than what was returned.
Document.count	Number of documents found.
Document[n].Library.id	Library identifier of the <i>n</i> th document.
Document[n].id	Document number of the <i>n</i> th document.
Document[n].version	Document version of the <i>n</i> th document.
Document[n].author	Author of the <i>n</i> th document.
Document[n].size	Size of the <i>n</i> th document.
Document[n].subject	Subject of the <i>n</i> th document.
Document[n].extension	Document file extension of the <i>n</i> th document.

Document[n].allowViewNative	1 enables native viewing for the document.
Document[n].allowView	1 enables HTML viewing for the document.
Document[n].Profile.createDate	Creation date of the <i>n</i> th document.
Document[n].Profile.modifyDate	Modification date of the <i>n</i> th document.
Document[n].Profile.officialVersion	Official version of the <i>n</i> th document.
Document[n].Profile.currentVersion	Current version of the <i>n</i> th document.
Document[n].Profile.versionCount	Number of versions for the <i>n</i> th document.

Example

```
<A href=/servlet/webpub?User.context=xyz&action=
Library.Search&Library.queryField=ALL&Library.queryId=
Lib1&Library.queryId=Lib2&Library.queryText=
WebAccess+Documents"> WebAccess Documents </A>
```

15 Revision History

The following table lists changes made to the GroupWise 8 WebAccess documentation:

November 2012	Reviewed and updated along with the GroupWise 2012 SDK guides.
October 2007	Changed the parameter descriptions of endsWith (page 54) to indicate that the strings are case sensitive.
June 2007	Added a note about using an UTF-8 compatible editor to edit the template files to Section 2.1, "Template Overview," on page 15 .
June 2006	Added the Item.Compose method to Compose.Action (page 115) .
March 2006	Added the Item.Accept.ShowAs parameter to Compose.Action (page 115) , Folder.Open (page 228) , Item.Accept (page 238) , and Item.Read (page 267) .
October 2005	Transitioned to revised Novell documentation standards.
June 2005	Added Item.MarkRead (page 253) .
March 2005	Added Compose.FolderSel (page 140) , Compose.SaveDraft (page 147) , and Item.Compose (page 249) . Added Item.Reply to Item.Read (page 267) . Added new Item.isDraft, Item.modified, Folder.hasDraft return values. Added Find.Source.draft parameter to Find.First (page 204) and the User.displayDraftItems parameter to User.Login (page 331) .
October 2004	Added XKI.getDRNList (page 278) to the Section 10.10, "Miscellaneous," on page 277 section of "Provider Methods" on page 105 .
June 2004	Added noScript (page 67) .
February 2004	Updated all figures. Edited entire document.
October 2003	Added descriptions for all the actions in the Find (page 201) , Section 10.14, "Send Options," on page 313 , Section 10.15, "Settings," on page 319 , Section 10.16, "Signature," on page 323 , and Section 10.17, "Time Zone," on page 326 . Added documentation for linkEncodedURLs (page 62) , xmlDecode (page 85) , formatMessage (page 59) , toWMLExt (page 81) , toQuote (page 77) , Compose.SendOptions.Modify (page 149) , Document.ViewNative (page 200) , Folder.ListChild (page 225) , Item.AcceptSharedAddressBook (page 240) , Item.CheckList.Bottom (page 243) , Item.CheckList.Down (page 244) , Item.CheckList.Save (page 245) , Item.CheckList.Top (page 246) , Item.CheckList.Up (page 247) , Item.Properties.Read (page 259) , Item.Properties.ReadInfo (page 262) , and Item.Properties.ReadNext (page 265) . Added about 25 values to the Section 6.1, "Configuration Settings," on page 87 table.

May 2002	<p>Corrected several minor errors in the text and in examples.</p> <p>Moved the section explaining template comments from the Template Keywords chapter to the Template Concepts chapter. See Section 2.4, "Comments," on page 18.</p> <p>Added reference documentation for the following new template keywords: Section 4.2, "ALIAS," on page 39, Section 4.3, "COOKIE," on page 41, Section 4.6, "HEADER," on page 42, Section 4.11, "NOOUTPUT," on page 44, and Section 4.12, "OUTPUT," on page 44.</p> <p>Changed the name of the "Template Methods" chapter to "Template Functions."</p> <p>Added some basic information about template functions at the beginning of Chapter 5, "Template Functions," on page 47.</p> <p>Added reference documentation for the following new template functions: getHeader (page 60), listToSortedArray (page 66), todayTZ (page 72), toDisplay (page 73), toHTMLExt (page 75), toScript (page 78), toXML (page 82), xmlEncode (page 86).</p> <p>Added reference documentation for the following GroupWise Provider Methods: Directory.Accumulate.Group (page 157), Directory.Accumulator.Add.BC (page 158), Directory.Accumulator.Add.CC (page 159), Directory.Accumulator.Add.To (page 160), Directory.Accumulator.Delete (page 161), Directory.Accumulator.List (page 162), Directory.CollapseGroup (page 165), Directory.CopyEntries (page 166), Directory.Create (page 167), Directory.Delete (page 174), Directory.GetLastSearch (page 176), Directory.HoldEntries (page 178), Directory.Info (page 179), Directory.InfoEntry (page 180), Directory.Modify (page 183), Directory.ModifyGroup (page 187), Directory.MoveEntries (page 189), Directory.SaveAs (page 190), Directory.ViewEntry (page 195), Directory.Cancel (page 214), Directory.GetRecipients (page 297), Folder.Action (page 217), Folder.Directory (page 220), Folder.ListShare (page 227), Folder.Share (page 231), Item.ReadLater (page 272), Directory.Cancel (page 280), Directory.GetRecipients (page 281), Proxy.Action (page 282), Proxy.AddAccess (page 283), Proxy.CloseListAccess (page 285), Proxy.CloseListHistory (page 286), Proxy.DeleteAccess (page 287), Proxy.DeleteHistory (page 288), Proxy.Directory (page 289), Proxy.ListAccess (page 290), Proxy.ListHistory (page 291), Proxy.Login (page 292), Proxy.ModifyAccess (page 293), Proxy.ReadAccess (page 295), Directory.GetRecipients (page 215), Rule.Action (page 298), Rule.Cancel (page 299), Rule.Create (page 300), Rule.CreateType (page 302), Rule.Delete (page 303), Rule.Directory (page 304), Rule.Enable (page 306), Rule.List (page 307), Rule.Modify (page 308), Rule.Read (page 310).</p> <p>Organized the GroupWise Provider Methods into topics according to their functional categories.</p>
February 2002	<p>Updated examples in the following: #String Table entry in Section 2.5, "Replaceable Parameters," on page 19, listToArray (page 65), Section 2.6, "Dates," on page 20, and Section 2.7, "Arrays," on page 20.</p>
September 2001	<p>Added support for GroupWise 6 to documentation.</p> <p>Added text alternatives for figures.</p>
June 2001	<p>Made minor format enhancements.</p> <p>Made changes improve document accessibility.</p>

July 2000	<p>Added the architecture overview graphics: GroupWise WebAccess Architecture and GroupWise WebPublisher Architecture.</p> <p>Added the following tasks sections: Section 3.3, "Modifying the Main Message Windows," on page 27, Section 3.4, "Modifying the Compose Windows," on page 29, Section 3.5, "Modifying the Address Book Windows," on page 30, Section 3.6, "Modifying the Calendar Windows," on page 31, and Section 3.7, "Modifying the Document Management Windows," on page 37.</p> <p>Documented two parameters for GroupWise Provider Method "Compose.Delegate" on page 136.</p>
May 2000	<p>Added the following overview sections: Section 1.2, "Changing the Company Logo," on page 12, Section 1.3, "Modifying the WebAccess User Interface," on page 13, and Section 1.4, "Adding Other Services or Databases," on page 13.</p> <p>Incorporated review comments from technical expert: adding new examples for <code>Item.ReadChild</code> and <code>Item.ReadParent</code>, adding new <code>addToDate</code> template method, fixing typos, and further clarifying the concepts outlined in this book.</p>
March 2000	<p>Rewrote and restructured documentation.</p>
