



Using the Breeze XML Web Services

Trademarks

ActiveEdit, ActiveTest, Add Life to the Web, Afterburner, Aftershock, Andromedia, Allaire, Animation PowerPack, Aria, Attain, Authorware, Authorware Star, Backstage, Blue Sky Software, Blue Sky, Breeze, Bright Tiger, Clustercats, ColdFusion, Contents Tab Composer, Contribute, Design In Motion, Director, Dream Templates, Dreamweaver, Drumbeat 2000, EDJE, EJIPT, Extreme 3D, Fireworks, Flash, FlashHelp, Flash Lite, FlashPaper, Flex, Flex Builder, Fontographer, FreeHand, Generator, Help To Source, HomeSite, Hotspot Studio, HTML Help Studio, JFusion, JRun, Kawa, Know Your Site, Knowledge Objects, Knowledge Stream, Knowledge Track, LikeMinds, Lingo, Live Effects, MacRecorder Logo and Design, Macromedia, Macromedia Action!, Macromedia Central, Macromedia Flash, Macromedia M Logo and Design, Macromedia Spectra, Macromedia xRes Logo and Design, MacroModel, Made with Macromedia, Made with Macromedia Logo and Design, MAGIC Logo and Design, Mediamaker, Movie Critic, Open Sesame!, RoboDemo, RoboEngine JFusion, RoboHelp, RoboHelp Office, RoboInfo, RoboInsight, RoboPDF, 1-Step RoboPDF, RoboFlash, RoboLinker, RoboScreenCapture, ReSize, Roundtrip, Roundtrip HTML, Shockwave, Sitespring, Smart Publishing Wizard, Software Video Camera, SoundEdit, Titlemaker, UltraDev, Web Design 101, what the web can be, WinHelp, WinHelp 2000, WinHelp BugHunter, WinHelp Find+, WinHelp Graphics Locator, WinHelp Hyperviewer, WinHelp Inspector, and Xtra are either registered trademarks or trademarks of Macromedia, Inc. and may be registered in the United States or in other jurisdictions including internationally. Other product names, logos, designs, titles, words, or phrases mentioned within this publication may be trademarks, service marks, or trade names of Macromedia, Inc. or other entities and may be registered in certain jurisdictions including internationally.

Third-Party Information

This guide contains links to third-party websites that are not under the control of Macromedia, and Macromedia is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Macromedia provides these links only as a convenience, and the inclusion of the link does not imply that Macromedia endorses or accepts any responsibility for the content on those third-party sites.

Copyright © 1997-2004 Macromedia, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc.

Acknowledgments

Project Management: Stephanie Gowin

Writing: Jody Bleyle

Managing Editor: Rosana Francescato

Editing: Mary Kraemer

Production Management: Patrice O'Neill

Media Design and Production: Adam Barnett, John Francis

Second Edition: September 2004

Macromedia, Inc.
600 Townsend St.
San Francisco, CA 94103

CONTENTS

INTRODUCTION: Before You Begin	9
Audience	9
Guide to instructional media	9
Additional resources	10
Typographical conventions	11
 CHAPTER 1: Architecture Overview	13
Data flow	14
Calling an action on the server	15
About parameters	15
About principals, SCOs, and IDs	16
About security	17
 CHAPTER 2: Working with Filters	21
About filters	21
About sort filters	23
Special filter scenarios	23
Filter reference	24
Testing code in the browser	24
Where to go from here	25
 CHAPTER 3: Common Tasks	27
Calling your first action	27
Integrating Breeze with a directory service	28
Integrating Breeze with a portal	32
Generating reports	35
 CHAPTER 4: Action Reference	37
Alphabetical action listing	37
Action reference entries	41
accesskey-exec	41
accesskey-info	42
acl-preference-update	43
action-list	46

common-info	47
custom-field-update.	47
custom-fields	48
custom-fields-delete.	49
group-membership-update	50
login	50
logout	52
permissions-info	52
permissions-reset	54
permissions-update	54
principal-info	55
principal-list	56
principal-update	57
principals-delete	59
report-account-meeting-attendance	59
report-active-meeting-presenters	61
report-active-meetings	61
report-bandwidth	62
report-bulk-content-quiz	63
report-bulk-content-quiz-results	64
report-bulk-content-slide-views	65
report-bulk-course-quiz	66
report-bulk-course-results	67
report-bulk-meeting	68
report-bulk-meeting-attendance	69
report-course-status	70
report-course-takers	71
report-disk-usage	72
report-meeting-attendance.	73
report-meeting-concurrent-users	74
report-meeting-session.	75
report-meeting-sessions	75
report-meeting-session-slots.	76
report-meeting-summary.	77
report-my-courses	78
report-my-meetings.	79
report-principal-list	80
report-quiz-answer-distribution.	81
report-quiz-definition-answers.	82
report-quiz-definition-questions	83
report-quiz-interactions.	84
report-quiz-question-answer-distribution	86
report-quiz-question-distribution	87
report-quiz-question-response	88
report-quiz-question-totals	89
report-quiz-summary.	90
report-quiz-takers	90
report-quotas.	92
report-sco-slides.	93
report-sco-views.	94

report-survey-question-response	94
report-training-concurrent-users	95
sco-build	96
sco-contents	97
sco-delete	98
sco-info	99
sco-move	101
sco-nav	101
sco-search	102
sco-shortcuts	103
sco-update	104
sco-upload	108
trees-info	109
user-accounts	110
user-update-pwd	111
 CHAPTER 5: XML Results Reference	 113
About returned XML code	113
Alphabetical list of XML tags	114
access-key (container)	119
access-key (key)	120
access-key-group	120
access-keys	121
account	121
action	122
actions	122
answer-correct	123
answer-text	123
body	123
comment	124
common	124
contact	125
cookie	125
course	126
custom-fields	126
date	127
date-begin	127
date-closed	128
date-created	128
date-end	128
date-expired	129
date-last-taken	129
date-modified	129
date-taken	130
date-time-attempted	130
description	131
domain-name	131
duration	131
email	132

expired	132
field.	132
first-name	133
hit	134
host	134
invalid	134
last-name.	135
last-viewed.	136
last-viewed-date.	136
login	136
meeting	137
my-courses	137
my-meetings	138
name	139
participant-name	139
permission.	139
permissions	140
physical-path	141
preferences.	141
presentation-name.	144
principal	144
principal-custom-field-values.	145
principal-list	146
principal-name	146
question	147
question-text	147
quiz-name	147
quiz-definition-answers	148
quiz-definition-questions.	148
quiz-name	149
quota.	149
report-account-meeting-attendance	150
report-active-meeting-presenters	151
report-active-meetings	151
report-bandwidth	152
report-bulk-content-quiz.	152
report-bulk-content-quiz-results	153
report-bulk-course-quiz	154
report-bulk-course-quiz-results	154
report-bulk-meeting	155
report-bulk-meeting-attendance	156
report-bulk-slide-views	156
report-course-status	157
report-disk-usage	158
report-meeting-attendance.	158
report-meeting-concurrent-users	159
report-meeting-session.	159
report-meeting-sessions	160
report-meeting-session-slots.	160
report-meeting-summary.	161

report-principal-list	161
report-quiz-answer-distribution	162
report-quiz-interactions	163
report-quiz-question-answer-distribution	163
report-quiz-question-distribution	164
report-quiz-question-response	165
report-quiz-question-totals	165
report-quiz-results	166
report-quiz-summary	166
report-quiz-takers	167
report-quotas	168
report-sco-slides	168
report-sco-views	169
report-survey-question-response	169
report-training-concurrent-users	170
response	170
results	171
row	171
sco	172
sco-author	172
sco-data	173
sco-name	173
sco-nav	174
sco-search-info	174
scos	175
shortcuts	176
source-sco	176
status	177
time-slot-begin	178
time-slot-end	179
time-taken	179
tree	179
trees	180
url	181
url-path	181
user (common-info)	181
user (user-accounts)	182
user-first-name	182
user-last-name	183
user-login	183
user-name	183
users	184
value	184
version	185
view-date-time	185
INDEX	187

INTRODUCTION

Before You Begin

The Macromedia Breeze XML web services enable your external system (such as a web application) to interact with a Breeze server.

This guide provides information on how to call XML web services (*actions*) on the Breeze server from your external system and how to interpret the XML results that Breeze returns. This guide also provides reference material explaining what each action does and what the parameters for each action are.

Note: To use the Breeze XML web services, you must have a valid access key. For more information, see [“About security” on page 17](#).

The web services discussed in this document are subject to change. Please see the updates page (www.macromedia.com/go/breeze_xmlapi_updates) for the latest updates and changes.

Audience

This book is intended for developers who want to integrate the Breeze XML web services into their external systems, such as web applications.

Before you use this book, you should understand the basics of XML and of using HTTP requests to communicate between an application and a server. You must also understand how to write code in a server language such as Macromedia ColdFusion Markup Language (CFML).

Guide to instructional media

Macromedia Breeze contains a variety of media to help you quickly learn how to use the product. In addition to this book, the following electronic manuals and online help systems are available:

- *Breeze Installation Guide* describes how to install the Breeze modules. This guide is available as a DVD insert for enterprise users, and as a PDF on the DVD.
- *Using the Macromedia Breeze Manager* describes how to use the administration, presentation, and training components of Breeze Live. You can access *Using the Breeze Manager* from the Breeze Manager Home page, and from the Help link in the Breeze Manager web application.

- *Breeze Live User Guide for Meeting Presenters* includes information about using the Breeze Live web application to host online real-time meetings. The documentation includes procedures that demonstrate the simplicity of adding slides, Flash SWF files, images, live audio, and video to your presentation. You can access *Breeze Live User Guide for Meeting Presenters* from the Breeze Manager Home page and from the Help menu within a Breeze Live meeting room (when you enter as a presenter) by selecting Meeting > Help.
- *Breeze Live User Guide for Meeting Participants* includes information relevant to users participating in an online real-time meeting. Breeze Live offers meeting participants several options to make their experience truly participatory, including the ability to send messages and questions to presenters and to each other, and the ability to participate through live video and audio. You can access *Breeze Live User Guide for Meeting Participants* from the Breeze Manager Home page and from the Help menu within a Breeze Live meeting room (when you enter as a participant) by selecting Meeting > Help.
- *Using the Macromedia Breeze Plug-In for Microsoft PowerPoint* describes use of the plug-in that allows users to create Breeze presentations from PowerPoint. You can access *Using the Breeze Plug-In for PowerPoint* from the Breeze Manager Home page and from the Breeze menu in PowerPoint by selecting Breeze > Help.
- *Using the Breeze Publish Wizard* includes procedures for publishing Breeze presentations created in PowerPoint to your Breeze server. You can access *Using the Breeze Publish Wizard* from the Breeze Manager Home page and from the Help button on the Publish Wizard.

Note: The relevance of these guides to users depends on the modules installed on the Breeze Presentation platform, the type of user (such as meeting content manager, information technology engineer, or course presenter), and the user's Breeze account permissions.

Additional resources

The following list contains some useful resources that are available on the web:

Note: For a list of Breeze resources, see “[Guide to instructional media](#)” on page 9.

The Web Service Primer at xml.com (<http://webservices.xml.com/pub/a/ws/2001/04/04/webservices/>) is a good introduction to web services.

“**Working with XML and ColdFusion**” in the *ColdFusion Developer's Handbook* (www.macromedia.com/devnet/mx/coldfusion/articles/cf_handbk_ch6.html) provides information about XML basics and creating XML code using CFML.

“**Leveraging XML with ColdFusion**” (www.macromedia.com/devnet/mx/coldfusion/articles/cf_handbk_ch7.html) discusses web services, Simple Object Access Protocol (SOAP), and how to call web services using CFML.

The XSL Transformations (XSLT) specification (www.w3.org/TR/xslt) introduces XSLT, which you can use to convert formatted data to other formats.

Numeric representation of dates and time (www.iso.ch/iso/en/prods-services/popstds/datesandtime.html) provides information about the ISO 8601 standard date and time format. More specifically, the W3C note about dates and times (www.w3.org/TR/NOTE-datetime) provides information about the specific profile of ISO 8601 that Breeze uses.

Typographical conventions

The following typographical conventions are used in this book:

- *Italic font* indicates a value that should be replaced (for example, in a folder path).
- `Code font` indicates code. It also indicates names of actions, names of parameters, names of tags, and names of attributes.
- **Bold font** indicates a verbatim entry.

CHAPTER 1

Architecture Overview

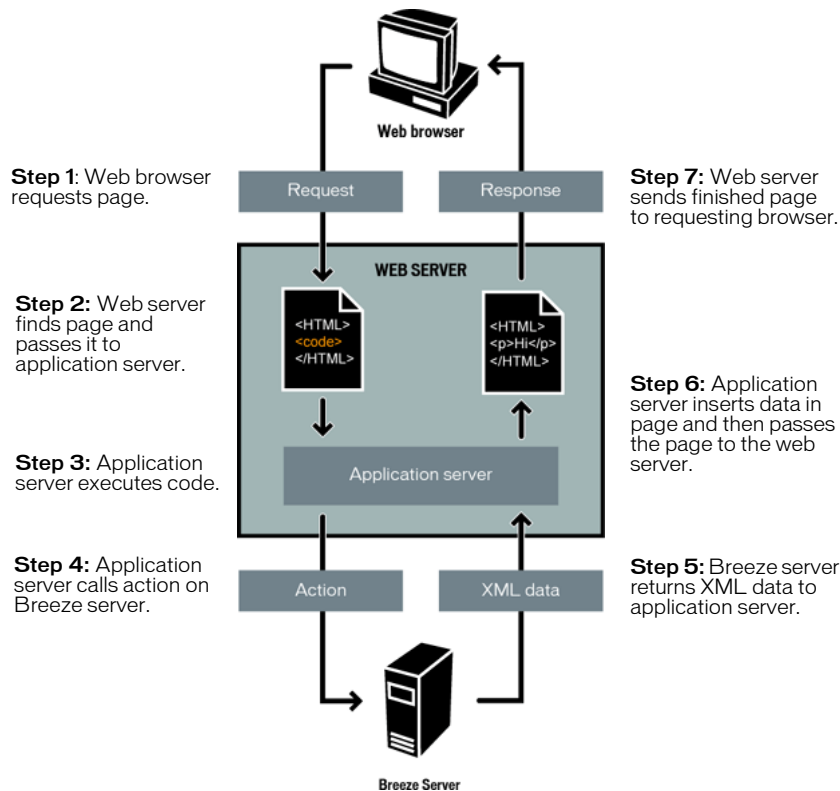
The Macromedia Breeze XML API model exposes interfaces as a set of XML web services. These services let your external system (such as a portal application) communicate with the Breeze server, using HTTP or HTTPS to call actions on the server and to receive results formatted as XML code. You can use the web services to add Breeze management and reporting capabilities to your external system.

Note: Breeze doesn't currently support Simple Object Access Protocol (SOAP).

This chapter describes the flow of data between the Breeze server and web applications and how you set parameters and security permissions.

Data flow

The following diagram shows an example of the flow of data in a web application that interacts with the Breeze server:



The following example describes what might happen when a user connects to a training portal intranet site that was created with Macromedia ColdFusion MX and that uses Breeze XML web services:

1. A logged-in user uses a web browser to request a page that shows a list of the courses that the user is signed up for.
2. The web server finds the relevant page and passes it to the application server.
3. The application server parses and executes the ColdFusion code contained in the page.
4. As part of executing the code, the application server calls an action on the company's Breeze server, requesting the list of courses. The action call takes the form of an HTTP `POST` request with appropriate security credentials included in the request.
5. The Breeze server executes the action, returning the resulting XML data to the application server.
6. The application code parses the returned XML and inserts data into the web page as appropriate, and then passes the finished page to the web server.
7. The web server passes the finished page to the browser to be displayed.

Calling an action on the server

To call an action on the Breeze server, pass the relevant parameters to the web services servlet at `http://server_name/api/xml`.

In Breeze 4.0, there are two parameters that are required for all actions: the parameter named `action` specifies the name of the action, and the parameter named `accesskey` specifies the account's access key. In Breeze 4.1, you do not need to pass the `accesskey` parameter. Passing the `accesskey` parameter in Breeze 4.1 is not harmful, it is ignored.

Most actions also have other parameters. For more information about access keys, see [“About security” on page 17](#). For more information about additional parameters for a given action, see the action's documentation in [Chapter 4, “Action Reference,” on page 37](#).

You can send the parameters to the server in either of the following ways:

- Create a query string—a URL that includes query parameters—and send it to the server as an HTTP GET or POST request, with the HTTP content type set to `application/x-www-form-urlencoded`. (In most cases, you don't need to explicitly set the content type when you use this approach.)

The URL to use for the query string has the form `http://breeze.example.com/api/xml?action=action_name&accesskey=access_key`, with the action's other parameters appended as additional query parameters separated by ampersands (&).

- Create an XML object containing the parameter data as `param` tags contained in a `params` tag, and then send an HTTP POST request to the server, with the HTTP content type set to `text/xml`.

For example code demonstrating how to call an action, see [Chapter 3, “Common Tasks,” on page 27](#).

Note: The example code in this book uses the query parameters approach, both for readability and because of limitations in using CFML to send XML objects containing parameter data. Macromedia recommends using the POST method when possible; because of the way HTTP implements GET and POST, using POST may provide higher security levels than passing parameters in the query string. For more information, see [“About security” on page 17](#).

About parameters

For most actions, you can specify one or more additional parameters. (The `accesskey` and `action` parameters are required. For more information, see [“Calling an action on the server” on page 15](#).)

You usually specify parameters as name-value pairs, but the exact process depends on the details of the language you use to call the action on the Breeze server. For sample code in ColdFusion Markup Language (CFML), see [Chapter 3, “Common Tasks,” on page 27](#).

In the documentation for each action, all listed parameters are required except for those specifically marked as optional.

For some actions, you can provide multiple parameters with the same name. For example, the [principals-delete](#) action takes a parameter named `principal-id`; you can specify multiple `principal-id` name-value pairs (each with the name `principal-id`) when you call the action on the Breeze server. The following CFML code example shows how to delete two specified users in a single action call, by specifying two `principal-id` parameters. (This example relies on variables called `baseurl`, `accesskey`, and `loginCookie` that you set during the login process. For more information, see [“Integrating Breeze with a directory service” on page 28.](#))

```
<cfset idOne=12345>
<cfset idTwo=23456>
<cfhttp url="#baseurl#api/xml?action=principals-
  delete&accesskey=#accesskey#&principal-id=#idOne#&principal-id=#idTwo#"
  method="post">
  <cfhttpparam type="Cookie" name="BREEZESSESSION" value="#loginCookie#">
</cfhttp>
```

In some other actions, you can provide multiple sets of parameters. For example, the [group-membership-update](#) action takes parameters named `group-id`, `principal-id`, and `is-member`; to make multiple updates in a single call, specify each of those parameters for principal and group, and then specify them (using the same parameter names again) for another, and so on.

About principals, SCOs, and IDs

There are two basic kinds of entities in Breeze: principals and Shareable Content Objects (SCOs).

A *principal* is any entity that can have permissions that control how it can interact with SCOs. The following are the most common types of principals:

`user` A specific user.

`group` A group of users.

There are other types of principals, known as *primary groups*: `admins`, `authors`, `live-admins`, and `course-admins`. For information about primary groups, see *Using the Breeze Manager*.

You can create and modify users and groups, but not primary groups.

A *SCO* is a Breeze document, such as a meeting or a presentation. Folders are also SCOs.

Breeze includes the following types of SCOs:

`tree` The root of a folder hierarchy. A tree's root is treated as an independent hierarchy; you can't determine the parent folder of a tree from inside the tree.

`folder` A folder on the server's hard disk, containing Breeze content.

`presentation` A presentation.

`course` A course.

`meeting` A meeting.

`flv` A Flash video file.

`swf` A Flash SWF file.

`image` An image file.

session A session of a meeting. A session begins when someone enters an empty meeting and ends when all attendees leave the meeting.

archive An archived meeting.

attachment A file shared using the file-sharing pod within a meeting.

pod A plug-in for a meeting.

logos An image (such as a corporate logo) that can be attached to content such as presentations.

Each principal and SCO has a unique ID number. When you call an action that uses an entity's ID as a parameter, the parameter name indicates what kind of entity the ID belongs to, but the ID remains the same, regardless of the parameter name. Parameter names for IDs include `acl-id`, `folder-id`, `group-id`, `parent-acl-id`, `principal-id`, `sco-id`, and `user-id`.

For example, to update a user's password, call the `user-update-pwd` action, and give the user's ID as the `user-id` parameter. To determine a user's name, call the `principal-info` action, and give the user's ID as the `principal-id` parameter.

There are a few other kinds of IDs, such as `account-id`, `answer-id`, `permission-id`, and `question-id`. But in most cases, a parameter name that ends in `-id` indicates that the parameter's value is either a principal's ID or a SCO's ID.

About security

The security model in Breeze ensures that any code calling a given action is authorized to do so. Almost every action call must include a cookie that represents a specific logged-in user. In Breeze 4.0, almost every action call must also include an access key.

In Breeze 4.1, you do not need to pass the `accesskey` parameter. However, passing the `accesskey` parameter in Breeze 4.1 is not harmful, it is ignored.

In Breeze 4.0, the access key (also called a *connection key*) is a code that prevents unauthorized use of the server. Every action (including `login`) requires an access key, which you specify as a parameter named `accesskey`. Access keys are case-sensitive. To locate your access key, see [“Access keys for Macromedia Breeze hosted \(ASP\) customers” on page 18](#) and [“Access keys for Macromedia Breeze Enterprise \(licensed\) customers” on page 18](#).

To call most actions, you must be acting as a particular logged-in user, so you must call the `login` action before you can perform most other actions. (The exceptions, which are the actions you can call without logging in, are [Action Reference](#), `action-list`, `common-info`, `login`, and `user-accounts`.)

When you log in, the Breeze server returns XML results, which indicate a successful login. The HTTP headers of those results include a cookie called `BREEZESSESSION`. When you receive the results of a `login` call, you must save the value of that cookie for later use. You must include that cookie's value in every subsequent request that you make as that logged-in user. For examples of how to do this, see [Chapter 3, “Common Tasks,” on page 27](#).

To successfully call an action, you must supply both an access key and the cookie for a logged-in user who has the permissions to call that action. If you don't supply both credentials (for example, if you supply the cookie for a user who doesn't have the right permissions), the action fails.

Your application can log in multiple users and can call actions for any of those users. For example, when an ordinary user is using your application, you may want to authenticate as that user and call actions using that user's credentials, while also keeping an administrative user logged in to perform tasks that require higher levels of permissions.

For information about the possible status codes that the server can return, see the [status](#) tag.

Access keys for Macromedia Breeze hosted (ASP) customers

Note: In Breeze 4.1, you no longer need an access key to call actions. However, passing the `accesskey` parameter in Breeze 4.1 is not harmful, it is ignored.

By default, your account is not enabled for XML web services. You must contact Macromedia technical support to obtain a valid XML access key to use with each action.

To contact technical support, go to the Breeze support page (www.macromedia.com/support/breeze/) and request that an XML access key be generated for your account.

Access keys for Macromedia Breeze Enterprise (licensed) customers

Note: In Breeze 4.1, you no longer need an access key to call actions. However, passing the `accesskey` parameter in Breeze 4.1 is not harmful, it is ignored.

Your server installation initially has no XML access key. You must add an access key before you can use the XML web services.

To add an access key:

1. Locate the `custom.ini` file, in the `breeze` subdirectory of the installation directory.
2. Add a parameter at the end of the file by typing the following text on a new line:
`XML_API_KEY=`
3. Immediately after the equal sign (=), on the same line, type an access key of your choice. The key should be 16 characters and should be a combination of numbers and letters. Access keys are case-sensitive.
4. Stop the Breeze server, and then restart it.

About permissions

Permissions define the ways in which a particular principal can interact with a given SCO.

A permission mapping, indicating what permissions a particular principal has for a particular SCO, is called an *ACL*. An ACL consists of three pieces of information: the ID of a SCO (usually referred to in this context as an ACL ID), the ID of a principal, and a keyword that indicates what the permissions are (usually referred to as a permission ID).

There are two kinds of permissions in Breeze: permissions associated with specific SCOs, and permissions that belong to all users who are members of special groups that are called primary groups. Permissions for primary groups take precedence over the permissions set on a SCO.

To find out what permissions a logged-in user has for a particular item, call the [permissions-info](#) action. To change a principal's permissions, call the [principal-update](#) action.

If a particular principal has no explicitly specified permissions on a particular SCO, that principal's permissions on the SCO's parent apply.

You can specify the following permissions on a SCO:

view The principal can view the SCO but not modify it. For a course, the **view** permission lets the principal enroll in the course. For a meeting, **view** lets the principal attend the meeting. For a folder, **view** lets the principal view the contents of the folder.

publish The principal can publish the SCO to the server and can update the SCO. This permission includes the **view** permission. It also lets the principal view reports related to the SCO. For a folder, **publish** doesn't let the principal create new folders within the folder or to set permissions for the folder. You cannot specify this permission on meetings or courses.

manage The principal can view, delete, move, and edit the SCO. This permission also lets the principal set permissions for the SCO. For a folder, the **manage** permission lets the principal view reports for files in the folder and create new folders. You cannot specify this permission on meetings or courses.

denied The principal cannot view, access, or manage the SCO. You cannot specify this permission on meetings or courses.

presenter (For meetings only) The principal can create or present a meeting. This permission lets the principal create or present the meeting, even if the principal doesn't have **view** permission on the meeting's parent folder.

Because a group is a principal, you can set these permissions on a SCO for a nonprimary group as well as for an individual user; if a group has a particular permission, all members of the group have that permission. Use the [group-membership-update](#) action to add a member to a group. Use the [permissions-update](#) action to set a group's permissions for a particular SCO.

The only way to obtain certain other kinds of permissions is to be a member of a primary group, as follows:

The account authors primary group lets its members create or edit a presentation anywhere in the system, regardless of the ACL permissions.

The account administrators primary group can manage users and groups.

The account meeting administrators primary group can create and present meetings anywhere in the system, regardless of the ACL permissions.

The course managers primary group can create and edit courses anywhere in the system, regardless of the ACL permissions.

To have full access to the entire system, you must be a member of all four primary groups.

About public access permissions

There is a special principal ID which, instead of being a number, has the value "public-access". This ID sets the default access setting for everyone, whether they are logged in or not. You can assign any of the following permissions on a SCO for the public-access principal:

`denied` Nobody can view, access, or manage the SCO.

`view` Anyone can view the SCO, even if not logged in.

`view-only` (For presentations only) Anyone can view the presentation, even if not logged in. However, the permissions set on the presentation's parent folder do not apply to the presentation. For example, even if a user has `manage` permission on the parent folder, the user can't delete a presentation that has `view-only` permission. (Normal permissions still apply to the presentation; if the user has `manage` permission on the presentation, the user can delete it.)

`view-hidden` (For meetings only) Anyone can attend the meeting, even if not logged in. However, the permissions set on the meeting's parent folder do not apply to the meeting.

Never assign `manage`, `presenter`, or `publish` permissions to the public-access principal. Never assign `view-only` or `view-hidden` permissions to normal principals.

About security and launching content

When you launch a SCO, you must provide authentication. You can do so using any of the following approaches:

- When you open the URL of the content, add a query parameter named `session` with a value equal to the value of the login cookie, as shown in the following example:

`http://breeze.example.com/p12345678/?session=breez3238uf298`

This approach is a potential security problem because anyone who obtains the specified URL can act as the logged-in user. If you take this approach, use the cookie for an ordinary user rather than the cookie for an administrative user.

Also, if a user gives the URL to someone else (for example, by copying it and pasting it into an e-mail message), they are giving access to their account, which presents a security risk.

- You can set a `BREEZESSESSION` cookie on the user's browser, using the value of the login cookie.

However, this approach works only if your application is running on a server with the same domain name as the Breeze server.

Also, if your application server is a J2EE servlet environment (such as ColdFusion or Java), the application server might also use a cookie named `BREEZESSESSION`, which results in potential conflicts between Breeze and the application server.

- You can simply open the URL, and require the user to log in again.

This approach is more secure than the others but can result in some inconvenience for users.

CHAPTER 2

Working with Filters

The data and information in your company's Macromedia Breeze content repository can grow significantly over time. When this occurs, you might not want to list every item in the repository for all users. For example, you could list the most recently created courses and the latest quarterly financial results presentation on your company's intranet. You need to sort and organize your growing repository to make sure users find the information they are looking for quickly.

Filters are the mechanism within Macromedia Breeze XML web services that you can use to define the criteria for retrieving data from Breeze. You use one or more filters with a specific XML API to ensure the data your users see matches exactly what they are looking for.

About filters

You can filter for many (but not all) fields in the XML, and you can filter to include values for that field or exclude values for that field. You can also sort the results and filter the return set to include less results, for example, the first 25 rows that match.

Filters work with action calls to modify or organize the data that the Breeze server returns. Filters help you select data, exclude data, and even sort the data you need to see. For example, you can request all your courses or use a filter to select only courses that include the word "Java"; you can list all users or select only users with the last name Smith and sort by their login name.

To use a filter, append it to an action call with an ampersand (&), as in the following:

```
report-my-course&filter-like-name=Java
```

You can string multiple filters together, each separated by an ampersand (&), as in the following:

```
report-my-course&filter-like-name=Java&sort-date-begin=desc
```

As you can see, filters are comprised of a *type* (either *filter* or *sort*) with an optional *modifier* (for example, *like*), the name of the *field* (for example, *name*), if needed, and finally, the *value* of the field name (for example, *Java*). The following is the filter format:

```
Type-Modifier-FieldName=Value
```

There are two types of filters, a basic filter, which begins with the word *filter*, and a sort filter, which begins with the word *sort*. You can combine sort filters and regular filters as needed to achieve your desired results. For more information about sort filters, see ["About sort filters"](#) on page 23.

Filter examples

The following is a simple example to help illustrate the general concept of filters. The web service API `report-my-courses` returns the list of courses you are enrolled in. This same API used with a filter on the course name, for example `report-my-courses&filter-like-name=Java`, retrieves your courses with the specified text in the name, in this case “Java”. If you add a sort filter such as `report-my-course&filter-like-name=Java&sort-date-begin=desc`, Breeze sorts your Java courses and displays the course you started first.

In these examples, there are two types of filters: filter and sort. The basic filter uses the `like` modifier on the `name` field with a value `Java`. If you omit the `like` modifier, Breeze would return courses with the exact name `Java` instead of courses with `Java` in the name. Using the sort filter without a modifier on the `date-begin` field with a value `desc` specifies a descending order.

The following table lists a few more examples that are helpful for understanding filter basics:

Filter	Description
<code>filter-name=Goals Review</code>	Returns items from the server that have the specific name “Goals Review.” This may apply to meetings, courses, or any items with a name field.
<code>filter-like-name=Goals</code>	Returns all items with “Goals” in the name such as “Goals Review” or “Quarterly Goals.”
<code>filter-out-name=Status</code>	Filters out or excludes any items with the word “Status” in the name field.
<code>filter-like-name=Goals &filter-out-status=active</code>	Returns all items with “Goals” in the name that are no longer active; for example, to find previous presentations on “Goals” from previous quarters.
<code>filter-gt-date-begin=2004-05-01&sort-name=asc</code>	Returns all items with a begin date later than May 1, 2004, sorted by name in ascending order.
<code>filter-gt-date-begin=2004-05-01&filter-lt-date-begin=2004-05-31</code>	Returns all items started in the month of May, 2004.

About date formatting

In Breeze, dates are a special type of field value that follow the ISO 8601 format. For example, May 28, 2004 is expressed as `2004-05-28` and 4:23pm on May 28, 2004 is expressed `2004-05-28T16:23:00.000`. Starting with the year, you may use as much of the date as you need to filter the best result. Using a value of `2004` returns all items that match the same year, which is probably not a good use of filters. You can retrieve everything that happened in a month by specifying the year and month. You add as much of the date as you need to select the most accurate data from the server.

About sort filters

You can use a filter of the type `sort` to sort data in ascending and descending order. For example, the following filter sorts the `name` field in ascending order:

```
sort-name=asc
```

The following code sorts the `name` field in descending order:

```
sort-name=desc
```

You can also perform primary and secondary sorts. For example, when listing calling the `principal-list` action to list principals, you can do a primary sort on the `type` field and then a secondary sort on the `name` field (this way, all principals of a specific type are grouped together and sorted by name in each group).

You can do this by specifying the following parameters:

```
sort1-type=asc&sort2-name=desc
```

Special filter scenarios

The following scenarios for filtering data vary for each action, for specific information about sort and filter options for individual actions, see their entries in [Chapter 4, “Action Reference,” on page 37](#).

To filter data that has a type field:

You can filter out or display only entries with a combination of matching types. For example, the following filter displays only the `types` folder and presentation:

```
filter-type=folder&filter-type=presentation
```

The following example shows all types except folder and presentation:

```
filter-out-type=folder&filter-out-type=presentation
```

If the `type` field allows a `null` value, you might want to filter out entries with `null` and `non-null` values. The following example shows only types that are `null`:

```
filter-type=null
```

The following example shows only types that are not `null`:

```
filter-out-type=null
```

To filter based on the length of a record:

The following shows all records whose date modified is between July 1 and July 10:

```
filter-lt-date-modified=2004-07-10T10:00:00.000-07:00&filter-gt-date-modified=2004-07-1T10:00:00.000-07:00
```

To filter a specific number of entries starting at a specific entry:

You can use this technique to create pagination when there is too much data for one page. The following example shows 25 records starting at the 100th record:

```
filter-rows=25&filter-start=100
```

Filter reference

Filters are comprised of a *type* (either *filter* or *sort*) with an optional *modifier*, the name of the *field*, if needed, and finally, the *value* of the field name, as shown in the following example:

Type-Modifier-FieldName=Value

The following table lists all the filter types and modifiers you can use with Breeze:

Type	Modifier	Description
filter	(none)	Field must match value exactly.
filter	like	Field must contain value.
filter	out	Field cannot contain value.
filter	rows	Limits the return result to number of rows specified in value. Does not use the field name.
filter	start	Limits the return result to number of rows specified in value. Does not use the field name.
filter	gt	Selects all items greater than the value. Works only with dates.
filter	lt	Selects all items greater than the value. Works only with dates.
filter	gte	Selects all items greater than or equal to the value. Works only with dates.
filter	lte	Selects all items less than or equal to the value. Works only with dates.
sort	(none)	Sorts results. Value must be <code>asc</code> or <code>desc</code> .

Testing code in the browser

You can enter a test URL in the address field of a browser window and see the XML response from the server. It is a good idea to become comfortable with filters in the browser before you write code.

There are many nuances to filters that determine which types of filters you can use with certain API calls. The browser approach lets you quickly try combinations to see what works and what doesn't. If you try this outside the browser, you add the extra time of compiling, loading the result in your application server, and debugging to see if a filter works as you expect. Using a browser, on the other hand, is much faster and gives you immediate feedback.

For example, you could use the following URL and filter to find all courses with the word “filter” in the name and the word “date” in the course description. This example URL uses `breezedev.mycompany.com` as the server name and `myflrst@pp^` for the API access key. Make sure you log on to the server first, and then enter the following URL, substituting your server name for `breezedev.mycompany.com` and your API access key for `myflrst@pp^`:

```
http://breezedev.mycompany.com/api/xml?accesskey= myflrst@pp^&action=report-  
my-courses&filter-like-name=filter&filter-like-description=date
```

The result in the browser should look similar to the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<results>
  <status code="ok" />
  <my-courses>
    <course sco-id="2649026" status="active" type="course">
      <name>Understanding Filters Part 2</name>
      <description>Second course in the series, covering important topics such
as date and range filtering.</description>
      <url>admin.ibreeze.macromedia.com/p40583853/</url>
      <date-created>2004-05-26T17:51:40.840-07:00</date-created>
      <date-modified>2004-05-26T17:51:47.750-07:00</date-modified>
      <date-begin>2004-05-24T14:00:00.000-07:00</date-begin>
      <url-path>/p40583853/</url-path>
      <expired>>false</expired>
    </course>
  </my-courses>
</results>
```

If your filters are too stringent and no courses meet your criteria, the output looks like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<results>
  <status code="ok" />
  <my-meetings />
</results>
```

Finally, as you experiment with filters and APIs in the browser, you might get an unexpected response. For example, if you enter the wrong filter field name, which matches some other item in the database, you could receive an XML response that starts with the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<results>
  <status code="internal-error">
```

To be production-ready, develop your code to handle unexpected errors and situations. Make sure you check for unexpected status codes and inform your IT team when you encounter such a situation.

Where to go from here

The Macromedia Breeze Resource Center has an article called [Working with Filters](#) that describes how to build a sample application that uses filters to search and sort a list of courses. In the article, you learn how to use filters with Breeze web services by working through a simple example that retrieves the list of enrolled courses for a user and allows the user to search and sort the list. It also explains the different ways that filters help you optimize retrieved data for use in other enterprise systems.

CHAPTER 3

Common Tasks

This chapter describes common scenarios for integrating Macromedia Breeze with external applications or systems. These scenarios show how to accomplish several common tasks. To perform each task, you call one or more actions on the Breeze server and then parse the XML tags that the server returns. In some cases, you use returned information as parameter values for the next action.

This chapter also provides sample code demonstrating how to perform some of the tasks using Macromedia ColdFusion Markup Language (CFML).

Note: Not all the task descriptions include sample code. The syntax for calling an action is consistent for all actions, so you can use the sample code from tasks that include it as models for code to perform other tasks.

Each time you call an action on the Breeze server, you must include an access key and credentials indicating which logged-in user is performing the action. For more information about the Breeze security model, including information about access keys, see [“About security” on page 17](#).

For detailed information about each action and each XML tag, see [Chapter 4, “Action Reference,” on page 37](#) and [Chapter 5, “XML Results Reference,” on page 113](#).

Calling your first action

You can enter a test URL in the address field of a browser window and see the XML response from the server. You should be comfortable with actions in the browser before you write code. A good action to call is `action-list` because it returns a list of all the other actions you can call.

To call an action in a browser, you need to know the following:

- The domain of your breeze account.
For example, `breeze.yourcompany.com`.
- The name of the action you want to call.
For a complete list of actions, see [“Action Reference” on page 37](#).

- In Breeze 4.0, the access key for your Breeze account. In Breeze 4.1, you do not need an access key, but it's not harmful if you use it.

For information about where to locate your access key, see [“Access keys for Macromedia Breeze hosted \(ASP\) customers” on page 18](#) and [“Access keys for Macromedia Breeze Enterprise \(licensed\) customers” on page 18](#).

To call a Breeze action in a browser:

1. Open a browser.
2. Enter the address of the breeze server in the browser's address bar. Append the following `action` and `accesskey` parameters in a query string:

```
http://breeze.example.com/api/xml?action=action_name&accesskey=access_key
```

Note: You do not need to append the `accesskey` parameter in Breeze 4.1.

3. Press Enter (Windows) or Return (Macintosh).

The action returns its XML results in the browser window. A call to `action-list` returns the following example:

```
<results>
  <status code="ok"/>
  <actions>
    <action>accesskey-exec</action>
    <action>accesskey-info</action>
    <action>acl-preference-update</action>
    <action>action-list</action>
    <action>custom-field-update</action>
    <action>custom-fields</action>
    <action>custom-fields-delete</action>
    <action>custom-fields-order-update</action>
    ...the list of actions continues here...
  </actions>
</results>
```

This is the same XML that returns when you call the action in an application. You can also test filters in the browser. For more information, see [“Testing code in the browser” on page 24](#).

Integrating Breeze with a directory service

Suppose your organization uses a central repository of user information, such as Lightweight Directory Access Protocol (LDAP). Breeze lets you import or synchronize user information from such a directory service.

The following procedure describes the steps that your application should follow to perform this task. This procedure assumes that if the information provided by the directory service doesn't match the information provided by the Breeze server, the information from the directory service is correct and up-to-date.

To synchronize Breeze with the directory service:

1. Log in by calling the `login` action on the Breeze server, specifying the login name and password of an account administrator. Examine the returned HTTP headers to find the value of the `BREEZESSESSION` cookie, which you need when calling subsequent actions.

Note: Consider creating an administrative user specifically for performing synchronizations. If you create such a user, you should exclude it from the list of users to synchronize when you perform the synchronization.

The following code example logs in a user and obtains the cookie value:

```
<cfset login="john@example.com">
<cfset password="abcdefg">
<cfset accesskey="0123456789101112">
<cfset baseurl="http://breeze.example.com/">

<cfset params="accesskey=#accesskey#&action=login&login=#login#&password=
#password#">
<cfhttp url="#baseurl#api/xml?#params#" method="get">

<cfset loginHeaders= "#cfhttp.ResponseHeader#" />

<cfloop collection="#loginHeaders#" item="httpHeader">
  <cfif httpHeader eq "Set-Cookie">
    <cfset value= loginHeaders[httpHeader] />
    <cfset cookieHeader="#value[1]#" />
    <cfloop list="#cookieHeader#" index="c" delimiters=";">
      <cfset cookieName= Left(c, Find('=', c)-1) />
      <cfif cookieName eq "BREEZESSESSION">
        <cfset fullCookie= mid(c, len(cookieName)+2,
          len(c)-len(cookieName)+1) />
        <cfset loginCookie= Left(fullCookie, Find('.', fullCookie)-1) />
      </cfif>
    </cfloop>
  </cfif>
</cfloop>
```

2. Check that the `login` action completed successfully by parsing the returned XML and examining the value of the `status` tag's `code` attribute. If the status code isn't "ok", determine what the error was and handle it appropriately.

```
<cfset xml= XmlParse(cfhttp.FileContent)>
<cfset resultsElement= XmlSearch(xml, "//status[@code='ok']")>
<cfif ArrayLen(resultsElement) ne 1>
  <!-- No status code "ok" found. Check for other status codes and handle
  errors. -->
</cfif>
```

3. Request a list of Breeze users from the Breeze server by calling the `principal-list` action. The server returns a complete list of all principals, including principals that aren't users. (For information about principals, see [“About principals, SCOs, and IDs” on page 16.](#))

The following code calls the `principal-list` action. The cookie value from the previous step is included as an HTTP parameter.

```
<cfhttp url="#baseurl#api/xml?action=principal-list&accesskey=#accesskey#"
  method="post">
  <cfhttpparam type="Cookie" name="BREEZESSESSION" value="#loginCookie#">
</cfhttp>
```

Note: This HTTP request uses the `"post"` method rather than `"get"` because the only way to specify an HTTP parameter in CFML is by using `"post"`. The URL and the query parameters remain the same as if you were using `"get"`, however.

4. Ensure that the action completed successfully by parsing the returned XML to ensure that the value of the `status` tag's `code` attribute is set to `"ok"`. If your code must be robust, you should check the `status` tag in the returned XML after every action.
5. If the status code is `"ok"`, search the returned list of principals to find all principals that have a `type` attribute set to `user`. The following example places a list of all Breeze users into the array `allUsers`:

```
<cfset xmlPrincipalList= XmlParse(cfhttp.FileContent)>
<cfset allUsers = XmlSearch(xmlPrincipalList, "//principal [@type='user']")>
```

6. Obtain a list of all the users listed in the directory service. The procedure for obtaining this list depends on the directory service.
7. Compare the values contained in the `login` tags for the Breeze users with the e-mail addresses from the directory service, using a list-comparison algorithm of your choice. Determine which users are in the directory service but not in Breeze; these are new users to be added to Breeze. Also determine which users are in Breeze but not in the directory service; these users should be removed from Breeze. Finally, determine which users are in both Breeze and the directory service, but have different names in the two places; these users' names should be changed in Breeze.

8. For each user listed in the directory service but not in Breeze, add the user to Breeze, as follows:

- a Obtain the user's first name, last name, and login name from the directory service.

The Breeze login name is usually the user's e-mail address.

- b Call the `principal-update` action, setting the `type` parameter equal to `user`.

Don't specify a `principal-id` parameter; leaving out `principal-id` indicates that you are adding a new user rather than updating an existing user.

- c Check the returned XML to ensure that the status code returned is "ok".

The following code demonstrates how to add a user to Breeze. It uses variables set up in previous steps of this procedure, such as `baseurl`. To keep the example clear and simple, this code specifies the user's information in the first few lines, rather than acquiring that information from a directory service.

```
<cfset newlogin= "jake2@example.com">
<cfset newpass= "abcdefg">
<cfset newfirst= "Jake">
<cfset newlast= "Doe">
<cfhttp url="#baseurl#api/xml?
    action=principal-update&accesskey=#accesskey#&first-name=#newfirst#
    &last-name=#newlast#&has-children=0&login=#newlogin#
    &type=user" method="post">
    <cfhttpparam type="Cookie" name="BREEZESESSION" value="#loginCookie#">
</cfhttp>
<!--- Check for an okay status result. --->
<cfset xml= XmlParse(cfhttp.FileContent)>
<cfset resultsElement= XmlSearch(xml, "//status[@code='ok']")>
<cfif ArrayLen(resultsElement) ne 1>
    <!--- No status code "ok" found. Check for other status codes and handle
        errors. --->
</cfif>
```

9. For each user listed in Breeze but not in the directory service, delete the user from Breeze by calling the `principals-delete` action and specifying the user's ID. Check the returned XML to ensure that the status code returned is "ok".

The following code demonstrates how to delete a specified user:

```
<cfset userID= "503123">
<cfhttp url="#baseurl#api/xml?
    action=principals-delete&accesskey=#accesskey#
    &principal-id=#userID#" method="post">
    <cfhttpparam type="Cookie" name="BREEZESESSION" value="#loginCookie#">
</cfhttp>
<!--- Check for an "ok" status code. --->
<cfset xml= XmlParse(cfhttp.FileContent)>
<cfset resultsElement= XmlSearch(xml, "//status[@code='ok']")>
<cfif ArrayLen(resultsElement) ne 1>
    <!--- No status code "ok" found. Check for other status codes and handle
        errors. --->
</cfif>
```

10. For each user whose information in Breeze must be updated, obtain the user's ID, as follows:
 - a Examine the data returned by the `principal-list` action in steps 2 through 4 and search for the ID associated with the user's login.
 - b Obtain the user's old first name and last name by calling the `principal-info` action.
 - c Change the values as needed to match the name given by the directory service.
 - d Call the `principal-update` action.
 - e Check the returned XML to ensure the status code returned is "ok".

The following code doesn't include code for calling `principal-update` or checking the status code because those procedures were described in earlier steps:

```
<cfset principal= XmlSearch(xmlPrincipalList,
    "//principal[login='#login#']">
<cfset principalId= principal[1].XmlAttributes["principal-id"]>

<cfhttp url="#baseurl#api/xml?
    action=principal-info&accesskey=#accesskey#&principal-id=#principalId#
    " method="post">
    <cfhttpparam type="Cookie" name="BREEZESESSION" value="#loginCookie#">
</cfhttp>

...
<!-- Add code here to check for an "ok" status code. -->
...
<cfset xml= XmlParse(cfhttp.FileContent)>
<cfset firstElement= XmlSearch(xml, "results/contact/first-name">
<cfset firstName= firstElement[1].XmlText>
<cfset lastElement= XmlSearch(xml, "results/contact/last-name">
<cfset lastName= lastElement[1].XmlText>
...
<!-- Add code here to set the new values for first name and last name,
    call principal-update, and check for an "ok" status code. -->
```

Integrating Breeze with a portal

Suppose you have a portal application, such as a company's intranet portal. You can integrate Breeze into the portal to let users view and modify Breeze-related information from within the context of the portal. For example, users can view upcoming Breeze meetings and see which courses they are enrolled in.

The procedures in this section describe how to perform various tasks for a portal application integrated with Breeze.

To log in as a particular user:

1. Obtain the user's login name and password, using a web-based form.
2. Call the `login` action on the Breeze server, using the specified login name and password.
3. Check the returned XML to ensure the status code returned is "ok".

To determine a user's ID, given their login name and password:

1. Log in as the user.
2. Call the `common-info` action.
3. Parse the returned XML to find the value of the `user-id` attribute of the `user` tag. That value is the user's ID.

To check whether an entered password is correct:

1. Log in as the user.
2. If the status code returned is "ok", the password is correct.

This procedure is the only way to check a password; for security reasons, Breeze doesn't let you request a user's password from the server.

To list all the courses the user is enrolled in:

1. Log in as the user.
2. Call the `report-my-courses` action.
3. Parse the returned XML and display the results. For more information about the returned XML, see [my-courses on page 137](#).

To list all the meetings the user has signed up to attend:

1. Log in as the user.
2. Call the `report-my-meetings` action.
3. Parse the returned XML and display the results. For more information about the returned XML, see [my-meetings on page 138](#).

To display the contents of the user's content folder:

1. Log in as the user.
2. Call the `sco-shortcuts` action.

This action provides the location of the root folder of the current account as well as the locations of other folders, such as the current user's meetings and other content.

The following code uses the same `baseUrl`, `accesskey`, and `loginCookie` variables that are set in the login code in ["Integrating Breeze with a directory service" on page 28](#):

```
<cfhttp url="#baseUrl#api/xml?action=sco-shortcuts&accesskey=#accesskey#"
method="post">
  <cfhttpparam type="Cookie" name="BREEZESESSION" value="#loginCookie#">
</cfhttp>
```

3. Determine the ID of the my-content folder, by searching for the appropriate type attribute in the returned `sco` tags:

```
<cfset response= cfhttp.FileContent >
<cfset xml= XmlParse(response)>
<cfset contentScoElement= XmlSearch(xml, "//sco[@type='my-content']")>
<cfset contentSco= contentScoElement[1].XmlAttributes["sco-id"]>
```

4. If you want to provide links to content items in the form of absolute URLs, determine the domain name of the folder:

```
<cfset domainElement= XmlSearch(xml,
    "//sco[@tree-id='#myTreeId#']/domain-name">
<cfset domainName= domainElement[1].XmlText>
```

5. Call the [sco-content](#)s action, using the folder's SCO ID.

The following action provides a list of the contents of the specified folder:

```
<cfhttp url="#baseUrl#api/xml?
    action=sco-content&accesskey=#accesskey#
    &sco-id=#contentSco#" method="post">
    <cfhttpparam type="Cookie" name="BREEZESSESSION" value="#loginCookie#">
</cfhttp>
```

6. Parse the returned XML and display relevant information to the user:

```
<cfset response= cfhttp.FileContent >
<cfset contentXml= XmlParse(response)>
<cfset contents= XmlSearch(contentXml, "//sco")>
<cfset contentCount= ArrayLen(contents)>
<cfoutput>
<table>
<tr>
    <th>SCO ID</th>
    <th>Name</th>
    <th>Type</th>
</tr>
<cfloop index="i" from="1" to="#contentCount#">
    <cfset item=contents[#i#]>
    <tr>
        <td>#item.XmlAttributes["sco-id"]#</td>
        <td>#item.name.XmlText#</td>
        <td>#item.XmlAttributes["type"]#</td>
    </tr>
</cfloop>
</table>
</cfoutput>
```

7. To provide links to launch content items, call the [sco-info](#) action for each item, and determine the item's full absolute URL by appending the returned `url-path` tag to the domain name you acquired. This results in a URL such as <http://breeze.example.com/p12345678/>.

For more information about launching content, see [“About security and launching content” on page 20](#).

To create a new meeting:

1. Log in as the user.
2. (Optional) Let the user determine the ID of the folder in which to create the new meeting, using a web-based interface. If you don't specify a folder, the meeting is created in the my-meetings folder. You can determine the ID of the my-meetings folder by calling the [sco-shortcuts](#) action.
3. Call the [sco-update](#) action, using the folder's ID for the `folder-id` parameter. Don't specify a `sco-id` parameter.
4. Verify that the action completed successfully by checking the returned `status` tag.

To search for content:

1. Obtain the string to search for from the user, using a web-based form.
2. Log in as the user.
3. Call the [sco-search](#) action, using the specified string as the value of the `query` parameter.
4. Parse the returned XML to find information about the SCOs that contain the string; display the relevant information to the user.

Generating reports

Suppose you want your application to provide information about all the courses a user is registered for or to identify the users who registered for a given meeting. The Breeze XML web services include a variety of reports, each of which provides data on Breeze use.

Report data is returned in XML form. You can use the results in your system, such as a web application, or you can convert the data into other file formats, such as comma-separated-values (CSV) files.

To generate a learner status report as a CSV file:

1. Log in as an administrator.
2. Determine the ID of a course. In most cases, you are presenting a list of courses (obtained by calling the [sco-contents](#) action, which also gives the IDs for the courses) in the application's UI, and the user selects a course.
3. Call the [report-course-takers](#) action.

This action returns a set of [row](#) tags, each providing information about a user who is signed up for the course.

4. Parse the XML and write the data as text, with commas separating the fields, to a CSV file. The most common approach shows the resulting CSV file as text in the user's browser, which lets the user save the file to their local disk.

You can use a server language such as CFML to translate the file to CSV, or you can use a transformation language such as XSLT. For information about XSLT, see [“Additional resources” on page 10](#).

CHAPTER 4

Action Reference

This chapter provides reference material for each action that is exposed in the Macromedia Breeze XML API, including information about the parameters that you can pass with each action. This chapter lists the actions in alphabetical order.

Every action returns a `results` tag that contains one or more other tags, including a `status` tag. The “Returned elements” section of each action’s documentation lists only the other elements returned; `results` and `status` are assumed.

For information about the XML tags, data, and status codes returned by the Breeze server, see [Chapter 5, “XML Results Reference,” on page 113](#).

For examples of how to perform various common tasks, see [Chapter 3, “Common Tasks,” on page 27](#).

For more information about specifying parameters, see [“About parameters” on page 15](#). For more information about principals and SCOs, see [“About principals, SCOs, and IDs” on page 16](#).

You must provide security credentials every time you call an action. For more information, see [“About security” on page 17](#).

Alphabetical action listing

The following list is all the actions documented in this reference chapter:

Action	Description
accesskey-exec	Executes special functions associated with special access keys.
accesskey-info	Provides a special access key, if such a key is associated with the specified SCO.
acl-preference-update	Updates the user profile to the specified language and time zone settings.
action-list	Returns a list of web service actions.
common-info	Provides basic information about the current user and server.

Action	Description
<code>custom-field-update</code>	Updates the specified account custom field. Creates a new one if none exists.
<code>custom-fields</code>	Lists the custom fields of an account and the details of the fields.
<code>custom-fields-delete</code>	Deletes the specified account custom field.
<code>group-membership-update</code>	Adds one or more principals to a group, or removes one or more principals from a group.
<code>login</code>	Logs in a user.
<code>logout</code>	Logs out a user, invalidating the cookie that the application received when the user logged in.
<code>permissions-info</code>	Provides information about principals and the permissions they have for a specified SCO.
<code>permissions-reset</code>	Resets all principals' permissions for the specified SCO, so the permissions of the SCO's parent apply to all principals.
<code>permissions-update</code>	Updates one or more principals' permissions for one or more SCOs.
<code>principal-info</code>	Provides information about the specified principal. The specified principal can be a user or a group.
<code>principal-list</code>	Provides a complete list of users and groups, including primary groups.
<code>principal-update</code>	Updates information for a principal in the current account or creates a new principal.
<code>principals-delete</code>	Deletes one or more principals.
<code>report-account-meeting-attendance</code>	Returns the meeting attendance log for the account.
<code>report-active-meeting-presenters</code>	Provides a list of the users who are currently presenting meetings.
<code>report-active-meetings</code>	Provides a list of meetings that are currently in progress.
<code>report-bandwidth</code>	Indicates the total bandwidth consumed in the current account since the account was created, in bytes.
<code>report-bulk-content-quiz</code>	Returns information about all content quizzes, including a list of quizzes, and the questions and answers for each quiz.
<code>report-bulk-content-quiz-results</code>	Returns results for a content quiz, including information on each user and quiz question.
<code>report-bulk-content-slide-views</code>	Returns slide view data for content.
<code>report-bulk-course-quiz</code>	Returns information about all course quizzes, including a list of quizzes, and the questions and answers for each quiz.
<code>report-bulk-course-results</code>	Returns results for course quiz, including information on each user and quiz question.
<code>report-bulk-meeting</code>	Returns information about all meetings.

Action	Description
report-bulk-meeting-attendance	Returns meeting attendance data.
report-course-status	Returns course status for either a principal or a SCO, based on the parameters passed in.
report-course-takers	Provides a list of the users enrolled in the specified course.
report-disk-usage	Provides information about how much hard disk space the Breeze content for the current account uses, in bytes.
report-meeting-attendance	Provides a list of users who have attended the specified meeting.
report-meeting-concurrent-users	Indicates the maximum number of users who can participate simultaneously in the specified meeting.
report-meeting-session	Provides information about a specific meeting session, such as the session name, session start and end times, and the number of participants and guests who attended the session.
report-meeting-sessions	Provides information about all the sessions of a meeting.
report-meeting-session-slots	Provides information about the number of attendees in every ten-minute time slot of the specified meeting session.
report-meeting-summary	Indicates how many users were invited to the specified meeting and how many invitees and guests attended.
report-my-courses	Provides information about each course that the logged-in user is enrolled in.
report-my-meetings	Provides information about each meeting the logged-in user is scheduled to attend.
report-principal-list	Returns the list of all principals in the account.
report-quiz-answer-distribution	Indicates how many users selected a particular answer to a particular quiz question.
report-quiz-definition-answers	Provides information about each of the allowed answers (both correct and incorrect) for the specified quiz.
report-quiz-definition-questions	Provides information about each question in a quiz.
report-quiz-interactions	Provides information about all the interactions that users have had with the specified quiz.
report-quiz-question-answer-distribution	Indicates how many users selected a specific answer for all questions on a quiz associated with the specified SCO.
report-quiz-question-distribution	Indicates how many users answered each question in the specified quiz correctly.
report-quiz-question-response	Provides a list of all answers that users have given to a particular quiz question.
report-quiz-question-totals	Indicates, for the specified quiz question, the total number of users who answered the question and the number of users who answered the question correctly.
report-quiz-summary	Provides information about the results of a quiz.

Action	Description
report-quiz-takers	Provides information about everyone who has taken a particular quiz.
report-quotas	Returns information on the account quotas.
report-sco-slides	Indicates how many times, and how recently, each slide in a presentation has been viewed.
report-sco-views	Indicates how many times, and how recently, the specified presentation has been viewed.
report-survey-question-response	Returns the list of all users who answered the specified question and their answer.
report-training-concurrent-users	Indicates the maximum number of concurrent course users for the specified account.
sco-build	Causes the Breeze server to build the specified SCO when you create presentations with the XML API.
sco-contents	Provides a list of the SCOs in a specified folder.
sco-delete	Deletes one or more SCOs.
sco-info	Provides information about a SCO.
sco-move	Moves a SCO from one folder to another.
sco-nav	Describes the folder hierarchy that contains the specified SCO.
sco-search	Provides a list of all SCOs that match the search text.
sco-shortcuts	Provides IDs for a set of folders containing content relevant to the logged-in user: a folder for the user to place meetings, a folder for the user to place content, and so on.
sco-update	Creates or updates presentations, courses, and meetings.
sco-upload	Uploads a file to the Breeze server when creating a presentation using the XML API. (For more information, see sco-update and sco-build).
trees-info	Returns the list of trees that belong to the specified account.
user-accounts	Provides a list of the accounts to which the specified user belongs.
user-update-pwd	Changes a user's password.

Action reference entries

The following section lists the Breeze web service actions alphabetically.

accesskey-exec

Description

Executes special functions associated with special access keys.

Currently, letting users self-register for meetings and courses is the only supported special function. The documentation for this action covers only the self-registration function.

To call this `accesskey-exec`, you need the normal access key for the account and a special access key associated with a meeting or course. Someone who isn't a Breeze user can be given the special access key and can then use a UI to register for the meeting or course, creating a Breeze user account for themselves in the process.

To obtain the special access key, call the [accesskey-info](#) action.

The difference between using `accesskey-exec` to create a new user and using [principal-update](#) is that you can call `accesskey-exec` without being logged in or being an administrator.

This action automatically adds the self-registering user to the self-registration group associated with the course or meeting, if the group exists. For more information, see [accesskey-info](#).

Additionally, you can specify custom fields for new users when you call `accesskey-exec`. To do so, pass the custom fields to `accesskey-exec` as additional parameters in name and value pairs, as shown in the following example:

```
field-id=xx&value=xxx&customfield-id=xx&value=xxx
```

The `field-id` parameters can be determined from the data returned by the [custom-fields](#) action.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze version 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`access-key` The special access key that lets the user self-register.

`first-name` The new user's first name.

`has-children` Always set this parameter to 0 when registering a new user.

`last-name` The new user's last name.

`login` The new user's login name, which is usually the user's e-mail address.

`password` The new user's password.

`type` Always set this parameter to "user" when registering a new user.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

accesskey-info

Description

Provides a special access key, if such a key is associated with the specified SCO.

For more information, see the [Action Reference](#) action.

To find out which group the user will automatically join by self-registering using the special access key, look in the `access-key-group` XML element. If the group exists, this element is included in the returned XML. If the group does not exist, there is no additional `access-key-group` element.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`acl-id` The ID of a SCO.

Filters

Results cannot be filtered or sorted.

Returned elements

[access-keys](#)

Sample results

The following sample XML is returned when a group doesn't exist and does not include an `access-key-group` element:

```
<results>
  <status code="ok" />
  <access-keys>
    <access-key acl-id="622847" action="action-self-reg-meeting"
      parent-acl-id="503265">
      <access-key>3hzvni fcfm phn3id</access-key>
      <date-created>2004-03-05T10:28:14.750-08:00</date-created>
    </access-key>
  </access-keys>
</results>
```

The following sample results are returned when a group exists and includes an accesskey-group element:

```
<results>
  <status code="ok" />
  <access-key-group account-id="7" has-children="true" is-primary="false"
    principal-id="630" type="group">
    <login>COURSE-test1-self-reg-group</login>
    <name>COURSE-test1-self-reg-group</name>
  </access-key-group>
  <access-keys>
    <access-key acl-id="126" action="action-self-reg-course" parent-acl-
      id="630">
      <access-key>2cw6np2kx2dtdop2</access-key>
      <date-created>2004-07-07T12:27:04.590-07:00</date-created>
    </access-key>
  </access-keys>
</results>
```

acl-preference-update

Description

Updates the user profile to the specified language and time zone settings.

Parameters

accesskey The access key for your account.

Note: The accesskey parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

acl-id The ID of the user for whom the preferences need to be updated.

lang The language setting. Breeze currently supports the following five languages:

Language	Parameter value
English	en
French	fr
German	de
Japanese	ja
Korean	ko

time-zone-id The time zone setting. Breeze currently supports the following 75 time zones:

Time zone setting	Parameter value
(GMT-12:00) International Date Line West	0
(GMT-11:00) Midway Island, Samoa	1
(GMT-10:00) Hawaii	2
(GMT-09:00) Alaska	3

Time zone setting	Parameter value
(GMT-08:00) Pacific Time (US and Canada); Tijuana	4
(GMT-07:00) Mountain Time (US and Canada)	10
(GMT-07:00) Chihuahua, La Paz, Mazatlan	13
(GMT-07:00) Arizona	15
(GMT-06:00) Central Time (US and Canada)	20
(GMT-06:00) Saskatchewan	25
(GMT-06:00) Guadalajara, Mexico City, Monterrey	30
(GMT-06:00) Central America	33
(GMT-05:00) Eastern Time (US and Canada)	35
(GMT-05:00) Indiana (East)	40
(GMT-05:00) Bogota, Lima, Quito	45
(GMT-04:00) Atlantic Time (Canada)	50
(GMT-04:00) Caracas, La Paz	55
(GMT-04:00) Santiago	56
(GMT-03:30) Newfoundland	60
(GMT-03:00) Brasilia	65
(GMT-03:00) Buenos Aires, Georgetown	70
(GMT-03:00) Greenland	73
(GMT-02:00) Mid-Atlantic	75
(GMT-01:00) Azores	80
(GMT-01:00) Cape Verde Islands	83
(GMT) Greenwich Mean Time Dublin, Edinburgh, Lisbon, London	85
(GMT) Casablanca, Monrovia	90
(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague	95
(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb	100
(GMT+01:00) Brussels, Copenhagen, Madrid, Paris	105
(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna	110
(GMT+01:00) West Central Africa	113
(GMT+02:00) Bucharest	115
(GMT+02:00) Cairo	120
(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius	125
(GMT+02:00) Athens, Istanbul, Minsk	130
(GMT+02:00) Jerusalem	135

Time zone setting	Parameter value
(GMT+02:00) Harare, Pretoria	140
(GMT+03:00) Moscow, St. Petersburg, Volgograd	145
(GMT+03:00) Kuwait, Riyadh	150
(GMT+03:00) Nairobi	155
(GMT+03:00) Baghdad	158
(GMT+03:30) Tehran	160
(GMT+04:00) Abu Dhabi, Muscat	165
(GMT+04:00) Baku, Tbilisi, Yerevan	170
(GMT+04:30) Kabul	175
(GMT+05:00) Ekaterinburg	180
(GMT+05:00) Islamabad, Karachi, Tashkent	185
(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi	190
(GMT+05:45) Kathmandu	193
(GMT+06:00) Astana, Dhaka	195
(GMT+06:00) Sri Jayawardenepura	200
(GMT+06:00) Almaty, Novosibirsk	201
(GMT+06:30) Rangoon	203
(GMT+07:00) Bangkok, Hanoi, Jakarta	205
(GMT+07:00) Krasnoyarsk	207
(GMT+08:00) Beijing, Chongqing, Hong Kong SAR, Urumqi	210
(GMT+08:00) Kuala Lumpur, Singapore	215
(GMT+08:00) Taipei	220
(GMT+08:00) Perth	225
(GMT+08:00) Irkutsk, Ulaan Bataar	227
(GMT+09:00) Seoul	230
(GMT+09:00) Osaka, Sapporo, Tokyo	235
(GMT+09:00) Yakutsk	240
(GMT+09:30) Darwin	245
(GMT+09:30) Adelaide	250
(GMT+10:00) Canberra, Melbourne, Sydney	255
(GMT+10:00) Brisbane	260
(GMT+10:00) Hobart	265
(GMT+10:00) Vladivostok	270

Time zone setting	Parameter value
(GMT+10:00) Guam, Port Moresby	275
(GMT+11:00) Magadan, Solomon Islands, New Caledonia	280
(GMT+12:00) Fiji Islands, Kamchatka, Marshall Islands	285
(GMT+12:00) Auckland, Wellington	290
(GMT+13:00) Nuku'alofa	300

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

action-list

Description

Returns a list of web service actions.

Caution: Not all actions in this list are documented, and not all documented actions are in this list.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

`actions`

Sample results

```
<results>
  <status code="ok" />
  <actions>
    <action>accesskey-exec</action>
    <action>accesskey-info</action>
    <action>action-list</action>
    ...
    [other actions listed here]
    ...
    <action>user-accounts</action>
```

```
<action>user-update-pwd</action>
</actions>
</results>
```

common-info

Description

Provides basic information about the current user and server.

If you call `common-info` without logging in first, the same information is returned, except that the `user` tag is not included, and the `account-id` value might be different.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

`common`

Sample results

```
<results>
  <status code="ok" />
  <common locale="en">
    <cookie>breezm5qtgnye46zpckbf.MARIANNE</cookie>
    <date>2004-02-12T15:53:19.797-07:00</date>
    <host>http://admin.breeze.example.com</host>
    <url>/api/xml?accesskey=yyyyyyyyyyyyyyyy&action=common-info</url>
    <version>breeze_402_r116</version>
    <account account-id="222914" />
    <user user-id="503562">
      <login>gerweck@example.com</login>
    </user>
  </common>
</results>
```

custom-field-update

Description

Updates the specified account custom field. Creates a new one if none exists.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

field-id The ID of the field to be updated (can be obtained by calling action [custom-fields](#)). This parameter needs to be specified only when updating an existing field.

name The name of the custom field (may be 1 to 60 characters long).

comments Any comments on the custom field (may be 0 to 60 characters long). This parameter is optional.

type The type of the custom field. Must be one of the following:

Value	Description
required	Required account custom field
optional	Optional and show during self-registration
optional-no-self-reg	Optional and hide during self-registration

custom-seq The custom field number. Specify this parameter only when creating a new custom field (should be one more than the current maximum value of **custom-seq**).

Returned elements

When updating an existing custom field: no elements are returned.

When creating a new custom field: [field](#).

Sample Results:

When creating a new custom field:

```
<results>
  <status code="ok" />
  <field field-id="1562" account-id="7" custom-seq="4"
    type="optional">
    <comments>none</comments>
    <name>Manager</name>
  </field>
</results>
```

When updating an existing custom field:

```
<results>
  <status code="ok" />
</results>
```

custom-fields

Description

Lists the custom fields of an account and the details of the fields.

Parameters

accesskey The access key for the account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results can be filtered only on `field-id`. You can also use the `filter-out` modifier on the `type` parameter.

Results cannot be sorted. The default sort is by ascending `field-id`.

Returned elements

`custom-fields`, `field`

Sample results

```
<results>
  <status code="ok" />
  <custom-fields>
    <field field-id="1329" custom-seq="1" account-id="7" type="optional">
      <name>Department</name>
      <comments>this is optional</comments>
    </field>
    <field field-id="1330" custom-seq="2" account-id="7"
type="required">
      <name>SSN</name>
    </field>
  </custom-fields>
</results>
```

custom-fields-delete

Description

Deletes the specified account custom field.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`field-id` The ID of the field to be deleted (call the `custom-fields` action to obtain the ID).

Returned elements

None.

Filters

Results cannot be filtered or sorted.

Sample results

```
<results>
  <status code="ok" />
</results>
```

group-membership-update

Description

Adds one or more principals to a group, or removes one or more principals from a group.

To update multiple principals and groups, specify multiple trios of `group-id`, `principal-id`, and `is-member` parameters. For more information, see [“About parameters” on page 15](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`group-id` The ID of the group for which you’re updating the membership.

`principal-id` The ID of the principal whose membership status you want to update.

`is-member` Indicates whether the principal is added to (`true`) or deleted from (`false`) the group.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

login

Description

Logs in a user.

After logging in, you must read and store the cookie called `BREEZESSESSION`, which can be found in the HTTP headers of the XML results. You must then include that cookie’s value in every subsequent request that you make as that logged-in user. For more information, see [“About security” on page 17](#).

The following example parses the HTTP headers to store the cookie and then indicates how it can be passed on for all subsequent requests.

To log in to Breeze and save the session cookie, use the following code:

```
login="john@example.com"
password="abcdefg"
accesskey="0123456789101112"
baseurl="http://breeze.example.com/"
```

```

URL loginUrl = new URL(baseUrl + "api/xml?accesskey=" + accessKey +
    "&action=login&login=" + login + "&password=" + password);

URLConnection conn=loginUrl.openConnection();
conn.connect();

InputStream resultStream=conn.getInputStream();
Document doc=new SAXBuilder(false).build(resultStream);

String cookieString=(String) (conn.getHeaderField("Set-Cookie"));

StringTokenizer st=new StringTokenizer(cookieString, "=");
if (st.countTokens() > 1 && st.nextToken().equals("BREEZESESSION")) {
    String cookieNext= st.nextToken();
    int semiIndex= cookieNext.indexOf(';');
    cookie= cookieNext.substring(0, semiIndex);
}

if (cookie == null){
    throw new RuntimeException("Couldn't find the Breeze cookie.");
}

```

To pass the BREEZESESSION cookie for all subsequent requests, use the following code:

```

URLConnection conn= url.openConnection();
conn.setRequestProperty("Cookie", "BREEZESESSION=" + cookie);
conn.connect();

```

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`account-id` The ID of the account that the user is associated with. This parameter is optional. If your organization has only one account, don't specify an `account-id`.

`login` The user's login name, which is usually the user's e-mail address.

`password` The user's password.

Note: If the `login` or `password` is missing or incorrect, the Breeze server returns a status code of "no-data".

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```

<results>
  <status code="ok" />
</results>

```

logout

Description

Logs out a user, invalidating the cookie that the application received when the user logged in.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

permissions-info

Description

Provides information about principals and the permissions they have for a specified SCO. You can also pass an optional `principal-id` parameter to retrieve permission information about a specific principal.

To determine the ID to use as the `principal-id` parameter, call the `principal-list` action (possibly with a filter to limit the returned data) to return an ID for the principal.

For more information about permissions, see [“About permissions” on page 18](#).

For more information about filters, see [Chapter 2, “Working with Filters,” on page 21](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`acl-id` The ID of a specific SCO.

`principal-id` The ID of a specific principal. This parameter is optional.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
is-primary	Filter and sort
type	Filter and sort
has-children	Filter and sort
permission-id	Filter and sort
name	Filter and sort
login	Filter and sort
description	Filter and sort

Note: Principals' logged-in-access and public-access are always returned in the data set, irrespective of the filter restrictions.

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[permissions](#)

Sample results

The following XML data is returned when the principal-id parameter is not passed and contains information about all principals:

```
<results>
  <status code="ok" />
  <permissions>
    <principal principal-id="181248" is-primary="false" type="user"
      has-children="false" permission-id="view">
      <name>Englesberg, Ari</name>
      <login>englesberg@example.com</login>
    </principal>
    <principal principal-id="181249" is-primary="false" type="user"
      has-children="false" permission-id="view">
      <name>Milligan, Susan</name>
      <login>milligan@example.com</login>
    </principal>
  </permissions>
</results>
```

The following XML data is returned when the principal-id parameter is passed:

```
<results>
  <status code="ok" />
  <permission acl-id="27" permission-id="manage" principal-id="8" />
</results>
```

If the principal has not been assigned permissions to the SCO, the status code no-data is returned.

permissions-reset

Description

Resets all principals' permissions for the specified SCO, so the permissions of the SCO's parent apply to all principals.

For information on permissions, see [“About permissions” on page 18](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`acl-id` The ID of a SCO.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

permissions-update

Description

Updates one or more principals' permissions for one or more SCOs.

For information about principals, see [“About principals, SCOs, and IDs” on page 16](#).

For information about permissions, see [“About permissions” on page 18](#).

To update multiple principals' permissions, specify multiple trios of `acl-id`, `permission-id`, and `principal-id` parameters. For more information, see [“About parameters” on page 15](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`acl-id` The ID of a SCO.

`permission-id` The ID of a permission.

`principal-id` The ID of a principal (a user or group).

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

principal-info

Description

Provides information about the specified principal. The specified principal can be a user or a group.

To determine the ID to use as the `principal-id` parameter, call the `principal-list` action (possibly with a filter to limit the returned data) to return an ID for the principal.

For information about principals, see [“About principals, SCOs, and IDs” on page 16](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`principal-id` The ID of a specific principal.

Filters

Results cannot be filtered or sorted.

Returned elements

`principal`, `contact`, `preferences`, `principal-custom-field-values`

Sample results

```
<results>
  <status code="ok" />
  <principal account-id="222914" has-children="true" is-primary="true"
    principal-id="222926" type="course-admins">
    <description>Course managers group</description>
    <login>Course Managers</login>
    <name>Course Managers</name>
  </principal>
</results>
```

The following sample XML data is returned when the `type` attribute of the `principal` element is `group`:

```
<results>
  <status code="ok" />
  <principal account-id="7" has-children="true" is-primary="false"
principal-id="122" type="group">
    <login>test-1</login>
    <name>test-1</name>
```

```

    </principal>
  </results>

```

When the type attribute of the principal element is user, additional data about the user contact information, user profile, and custom fields for the user is returned, as shown in the following XML data:

```

<results>
  <status code="ok" />
  <contact contact-id="654">
    <email>test4-lnagaraj@test.enang.com</email>
    <first-name>test4</first-name>
    <last-name>laxmi</last-name>
  </contact>
  <preferences acl-id="653" lang="en" time-zone-id="4" />
  <principal account-id="7" has-children="false" is-primary="false"
principal-id="653" type="user">
    <login>test4-lnagaraj@test.enang.com</login>
    <name>test4 laxmi</name>
  </principal>
  <principal-custom-field-values>
    <field field-id="652" custom-seq="1" account-id="7" type="optional"
principal-id="653">
      <name>Phone number</name>
      <value>408-1111-111</value>
    </field>
  </principal-custom-field-values>
</results>

```

See also

[principal-list](#), [principal-update](#)

principal-list

Description

Provides a complete list of users and groups, including primary groups.

Parameters

accesskey The access key for your account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
principal-list	Filter and sort
contact-id	Filter and sort
type	Filter and sort

Field	Filter/Sort
has-children	Filter and sort
is-primary	Filter and sort
is-hidden	Filter and sort
name	Filter and sort
login	Filter and sort
description	Filter and sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[principal-list](#)

Sample results

```
<results>
  <status code="ok" />
  <principal-list>
    <principal principal-id="8" contact-id="3" type="user" has-
children="false" is-primary="false" is-hidden="0">
      <name>Enterprise Administrator</name>
      <login>lnagaraj@macromedia.com</login>
      <email>lnagaraj@macromedia.com</email>
    </principal>
    <principal principal-id="18" contact-id="" type="authors" has-
children="true" is-primary="true" is-hidden="0">
      <name>Account Authors</name>
      <login>Account Authors</login>
      <description>Account authors group</description>
    </principal>
  </principal-list>
</results>
```

principal-update

Description

Updates information for a principal in the current account or creates a new principal. For information about principals, see [“About principals, SCOs, and IDs” on page 16](#).

In most cases, you must acquire the original version of the principal’s information (using [principal-info](#)) before calling the principal-update action.

To create a new principal, call the principal-update action without specifying a principal-id parameter. If you create a new principal, the returned XML provides information about the new principal, including the principal’s ID.

To specify custom field values when creating or updating a user, pass the custom field values as name, value pairs, as shown in the following example:

```
field-id=xx&value=xxx&field-id=xx&value=xxx
```

You can determine the `field-id` values from the data returned by the [custom-fields](#) action.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`description` The new group's description. Use only when creating a new group.

`first-name` The new first name to assign to the user. Use only when creating or updating a user.

`has-children` Indicates whether the principal has children. If the principal is a group, set this parameter to 1. If the principal is a user, set this parameter to 0.

`last-name` The new last name to assign to the user. Use only when creating or updating a user.

`login` The principal's new login name, which is usually the principal's e-mail address. This parameter must be specified only when creating or updating a user.

`name` The new group's name. Use only when creating a new group.

`password` The new user's password. Use only when creating a new user.

`principal-id` The ID of the principal for which you're changing data. If you omit this parameter, the `principal-update` action creates a new principal.

`type` The type of the new principal. Use only when creating a new principal. For information about the available types, see [“About principals, SCOs, and IDs” on page 16](#).

Filters

Results cannot be filtered or sorted.

Returned elements

If you update an existing principal, no elements are returned.

If you create a new principal, [principal](#) (in the same format as the returned elements for [principal-info](#)).

Sample results

The following sample results are for creating a new principal:

```
<results>
  <status code="ok" />
  <principal type="user" principal-id="2500848" has-children="0" account-
    id="181224">
    <login>jake3@example.com</login>
    <name>doe, jake</name>
  </principal>
</results>
```

See also

[principal-info](#), [principal-update](#), [principals-delete](#)

principals-delete

Description

Deletes one or more principals.

To delete multiple principals, specify multiple `principal-id` parameters. For more information, see [“About parameters” on page 15](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`principal-id` The ID of a principal to delete.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

report-account-meeting-attendance

Description

Returns the meeting attendance log for the account.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>tr.sco_id</code>	Filter and sort
<code>transcript-id</code>	Filter and sort
<code>principal-id</code>	Filter and sort

Field	Filter/Sort
login	Filter and sort
sco-name	Sort
date-created	Filter and sort
date-closed	Filter and sort
participant-name	Sort

Returned elements

[report-account-meeting-attendance](#)

Sample results

```
<results>
  <status code="ok" />
  <report-account-meeting-attendance>
    <row account-id="7" transcript-id="670" sco-id="623" principal-id="8"
      contact-id="3">
      <login>lnagaraj@macromedia.com</login>
      <session-name>Enterprise Administrator</session-
        name>
      <sco-name>test-mtg-1</sco-name>
      <date-created>2004-07-09T13:00:28.920-07:00</date-
        created>
      <date-closed>2004-07-09T13:02:55.810-07:00</date-
        closed>
      <participant-name>Enterprise Administrator</participant-
        name>
    </row>
    <row account-id="7" transcript-id="685" sco-id="623" principal-id="8"
      contact-id="3">
      <login>lnagaraj@macromedia.com</login>
      <session-name>Enterprise Administrator</session-
        name>
      <sco-name>test-mtg-1</sco-name>
      <date-created>2004-07-09T14:33:06.577-07:00</date-
        created>
      <date-closed>2004-07-09T14:33:43.547-07:00</date-
        closed>
      <participant-name>Enterprise Administrator</participant-
        name>
    </row>
  </report-account-meeting-attendance>
</results>
```

report-active-meeting-presenters

Description

Provides a list of the users who are currently presenting meetings.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a meeting; if specified, the report returns information about that meeting only. This parameter is optional.

Filters

Results cannot be filtered or sorted.

Returned elements

`report-active-meeting-presenters`

Sample results

```
<results>
  <status code="ok" />
  <report-active-meeting-presenters>
    <sco sco-id="190298">
      <name>Jack Monson</name>
      <date-created>2004-02-17T17:06:23.920-08:00</date-created>
    </sco>
    <sco sco-id="566058">
      <name>Hans Gay</name>
      <date-created>2004-02-17T16:50:00.327-08:00</date-created>
    </sco>
  </report-active-meeting-presenters>
</results>
```

report-active-meetings

Description

Provides a list of meetings that are currently in progress.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-active-meetings](#)

Sample results

```
<results>
  <status code="ok" />
  <report-active-meetings>
    <sco sco-id="196311" active-participants="1" length-minutes="60">
      <name>QT load</name>
      <date-begin>2003-09-02T14:00:58.330-07:00</date-begin>
    </sco>
    <sco sco-id="184847" active-participants="5" length-minutes="30">
      <name>Take 2</name>
      <date-begin>2003-09-02T14:45:25.273-07:00</date-begin>
    </sco>
  </report-active-meetings>
</results>
```

report-bandwidth

Description

Indicates the total bandwidth consumed in the current account since the account was created, in bytes.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-bandwidth](#)

Sample results

```
<results>
  <status code="ok" />
  <report-bandwidth total="71000000" />
</results>
```

report-bulk-content-quiz

Description

Returns information about all content quizzes, including a list of quizzes, and the questions and answers for each quiz. The information returned is similar to the “Content quiz Information” download report, except that XML data is returned instead of CSV formatted data.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

`report-bulk-content-quiz`

Sample results

```
<results>
  <status code="ok" />
  - <report-bulk-content-quiz>
    <row presentation-id="720" quiz-id="1" quiz-passing-score="10" question-
      number="1" question-value="10" answer-id="1">
      <presentation-name>test-quiz</presentation-name>
      <quiz-name>Quiz test 1</quiz-name>
      <question-text>Who is Randy Johnson ?</question-text>
      <answer-text>A baseball pitcher with the Arizona Diamondbacks</answer-
        text>
      <answer-correct>YES</answer-correct>
    </row>
    - <row presentation-id="720" quiz-id="1" quiz-passing-score="10" question-
      number="1" question-value="10" answer-id="2">
      <presentation-name>test-quiz</presentation-name>
      <quiz-name>Quiz test 1</quiz-name>
      <question-text>Who is Randy Johnson ?</question-text>
      <answer-text>A basketball player with the New Jersey Nets</answer-text>
      <answer-correct>NO</answer-correct>
    </row>
  </report-bulk-content-quiz>
</results>
```

report-bulk-content-quiz-results

Description

Returns results for a content quiz, including information on each user and quiz question. The returned information is similar to the “Content quiz responses and results” download report, except that the `report-bulk-quiz-results` action returns XML data instead of CSV formatted data.

Caution: There was an issue with the formatting of the `time-taken` XML tag, which has been fixed in the Breeze 4.1 updater. If you see incorrect formatting in the content of a `time-taken` XML tag, download the updater from the [Licensed Support Center](#).

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>user-id</code>	Filter and sort
<code>presentation-attempt-id</code>	Filter only
<code>presentation-id</code>	Filter only
<code>quiz-id</code>	Filter only
<code>question-number</code>	Filter and sort
<code>user-response</code>	Filter only
<code>user-login</code>	Filter only
<code>date-time-attempted</code>	Filter only

Returned elements

`report-bulk-content-quiz-results`

Sample results

```
<results>
  <status code="ok" />
  <report-bulk-content-quiz-results>
    <row user-id="8" presentation-attempt-id="725" presentation-id="720"
      quiz-id="1" question-number="1" user-response="1">
      <user-login>lnagaraj@macromedia.com</user-login>
      <date-time-attempted>2004-07-13T10:51:35.047-07:00</date-time-
        attempted>
      <time-taken>00:00:28.017</time-taken>
    </row>
    <row user-id="8" presentation-attempt-id="726" presentation-id="720"
      quiz-id="1" question-number="2" user-response="3">
```

```

        <user-login>lnagaraj@macromedia.com</user-login>
        <date-time-attempted>2004-07-13T10:52:23.780-07:00</date-time-
attempted>
        <time-taken>00:00:45.046</time-taken>
    </row>
</report-bulk-content-quiz-results>
</results>

```

report-bulk-content-slide-views

Description

Returns slide view data for content. This is similar to the “Content Slide Views” download report, except that the action returns XML data and the report returns CSV data.

Parameters

accesskey The access key for the account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
presentation-id	Sort
slide-number	Filter and sort
user-login	Sort
user-first-name	Sort
user-last-name	Sort
presentation-name	Sort
view-date-time	Sort

Returned elements

[report-bulk-slide-views](#)

Sample results

```

<results>
  <status code="ok" />
  <report-bulk-slide-views>
    <row session-id="10000022" viewing-session="32" presentation-id="27"
slide-number="1">
      <user-login>lnagaraj@macromedia.com</user-login>
      <user-first-name>Enterprise</user-first-name>
      <user-last-name>Administrator</user-last-name>
      <presentation-name>test-slide</presentation-name>
      <view-date-time>2004-06-30T11:46:10.280-07:00</view-date-time>
    </row>
  </report-bulk-slide-views>
</results>

```

```

        <row session-id="10000022" viewing-session="32" presentation-id="27"
slide-number="2">
            <user-login>lnagaraj@macromedia.com</user-login>
            <user-first-name>Enterprise</user-first-name>
            <user-last-name>Administrator</user-last-name>
            <presentation-name>test-slide</presentation-name>
            <view-date-time>2004-06-30T11:46:12.500-07:00</view-date-time>
        </row>
    </report-bulk-slide-views>
</results>

```

report-bulk-course-quiz

Description

Returns information about all course quizzes, including a list of quizzes, and the questions and answers for each quiz. The returned results are similar to the “Course Quiz Information” report, except that the action returns XML data and the report contains downloadable CSV data.

Parameters

accesskey The access key for the account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-bulk-course-quiz](#)

Sample results

```

<results>
  <status code="ok" />
  <report-bulk-course-quiz>
    <row course-id="727" quiz-id="1" quiz-passing-score="10" question-
number="1" question-value="10" answer-id="1">
      <presentation-name>quiz-course-1</presentation-name>
      <quiz-name>Quiz test 1</quiz-name>
      <question-text>Who is Randy Johnson ?</question-text>
      <answer-text>A baseball pitcher with the Arizona Diamondbacks</answer-
text>
      <answer-correct>YES</answer-correct>
    </row>
    <row course-id="727" quiz-id="1" quiz-passing-score="10" question-
number="1" question-value="10" answer-id="2">
      <presentation-name>quiz-course-1</presentation-name>
      <quiz-name>Quiz test 1</quiz-name>
      <question-text>Who is Randy Johnson ?</question-text>
      <answer-text>A basketball player with the New Jersey Nets</answer-text>
      <answer-correct>NO</answer-correct>
    </row>
  </report-bulk-course-quiz>
</results>

```

report-bulk-course-results

Description

Returns results for a course quiz, including information on each user and quiz question. The returned data is similar to the “Course Quiz Responses and Results” report, except that the action returns XML data and the report returns downloadable CSV data.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Fields	Filter/Sort
<code>user-id</code>	Filter and sort
<code>course-attempt-id</code>	Sort
<code>course-id</code>	Sort
<code>quiz-id</code>	Sort
<code>question-number</code>	Filter and sort
<code>user-response</code>	Sort
<code>user-login</code>	Sort
<code>date-time-attempted</code>	Sort
<code>time-taken</code>	Sort

Returned elements

[report-bulk-course-quiz-results](#)

Sample results

```
<results>
  <status code="ok" />
  <report-bulk-course-quiz-results>
    <row user-id="8" course-attempt-id="730" course-id="727" quiz-id="1"
      question-number="1" user-response="2">
      <user-login>lnagaraj@macromedia.com</user-login>
      <date-time-attempted>2004-07-13T10:54:09.030-07:00</date-time-
        attempted>
      <time-taken>00:00:14.017</time-taken>
    </row>
    <row user-id="8" course-attempt-id="730" course-id="727" quiz-id="1"
      question-number="2" user-response="1">
      <user-login>lnagaraj@macromedia.com</user-login>
      <date-time-attempted>2004-07-13T10:54:09.030-07:00</date-time-
        attempted>
```

```

        <time-taken>00:00:34.017</time-taken>
    </row>
</report-bulk-course-quiz-results>
</results>

```

Caution: There was an issue with the formatting of the `time-taken` XML tag, which has been fixed in the Breeze 4.1 updater. If you see incorrect formatting in the content of a `time-taken` XML tag, download the updater from the [Licensed Support Center](#).

report-bulk-meeting

Description

Returns information about all meetings. Similar to the Meeting Information report, except that the action returns XML data and the report returns downloadable CSV data.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-bulk-meeting](#)

Sample results

```

<results>
  <status code="ok" />
  <report-bulk-meeting>
    <row meeting-id="623" invited-participants="3">
      <meeting-name>test-mtg-1</meeting-name>
      <meeting-scheduled-date-time>2004-07-07T10:00:00.000-07:00</meeting-
scheduled-date-time>
    </row>
    <row meeting-id="679" invited-participants="2">
      <meeting-name>test-mtg-2</meeting-name>
      <meeting-scheduled-date-time>2004-07-09T14:00:00.000-07:00</meeting-
scheduled-date-time>
    </row>
  </report-bulk-meeting>
</results>

```

report-bulk-meeting-attendance

Description

Returns meeting attendance data. The results are similar to the “Meeting Attendance” report, except that the action returns XML data and the report returns downloadable CSV data.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
meeting-id	Sort
meeting-attendee	Sort
meeting-name	Sort
attendee	Sort
attendee-join	Sort
attendee-exit	Sort

Returned elements

[report-bulk-meeting-attendance](#)

Sample results

```
<results>
  <status code="ok" />
  <report-bulk-meeting-attendance>
    <row meeting-id="623" meeting-attendee="8">
      <meeting-name>test-mtg-1</meeting-name>
      <attendee>Enterprise Administrator</attendee>
      <attendee-join>2004-07-09T13:00:28.920-07:00</attendee-join>
      <attendee-exit>2004-07-09T13:02:55.810-07:00</attendee-exit>
    </row>
    <row meeting-id="623" meeting-attendee="8">
      <meeting-name>test-mtg-1</meeting-name>
      <attendee>Enterprise Administrator</attendee>
      <attendee-join>2004-07-09T14:33:06.577-07:00</attendee-join>
      <attendee-exit>2004-07-09T14:33:43.547-07:00</attendee-exit>
    </row>
  </report-bulk-meeting-attendance>
</results>
```

report-course-status

Description

Returns course status for either a principal or a SCO, based on the parameters passed in.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`principal-id` The ID of the principal for whom the courses status is requested.

`sco-id` The ID of the SCO for which the course status is requested.

Note: Only one of the `principal-id` or `sco-id` parameters needs to be passed in based on the type of information needed.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-course-status](#)

Sample results

When the `sco-id` parameter is passed in, the following XML may be returned:

```
<results>
  <status code="ok" />
  <report-course-status total-course-completions="2" total-unique-course-
    completions="1" num-passed="1" num-failed="1" num-enrollees="3">
    <date-last-taken>2004-07-13T10:55:28.763-07:00</date-last-taken>
  </report-course-status>
</results>
```

When the `principal-id` parameter is passed in, the following XML may be returned:

```
<results>
  <status code="ok" />
  <report-course-status total-course-completions="4" total-unique-course-
    completions="4" num-passed="3" num-failed="1">
    <date-last-taken>2004-07-27T12:53:42.297-07:00</date-last-taken>
  </report-course-status>
</results>
```

report-course-takers

Description

Provides a list of the users enrolled in the specified course.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a course.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>transcript-id</code>	Filter and sort
<code>sco-id</code>	Filter and sort
<code>status</code>	Filter and sort
<code>percentage-score</code>	Filter and sort
<code>time-taken</code>	Sort
<code>certificate</code>	Sort
<code>answered-survey</code>	Sort
<code>login</code>	Sort
<code>date-taken</code>	Filter and sort
<code>principal-name</code>	Sort
The data can be sorted on any of the custom fields that exist for the account.	Sort

You cannot filter on any custom field.

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[report-quiz-takers](#)

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-takers>
    <row transcript-id="" sco-id="" principal-id="8" status="" score=""
percentage-score="" time-taken="" certificate="" answered-survey="0">
      <login>lnagaraj@macromedia.com</login>
      <principal-name>Enterprise Administrator</principal-name>
      <custom0 />
    </row>
    <row transcript-id="" sco-id="" principal-id="123" status="" score=""
percentage-score="" time-taken="" certificate="" answered-survey="0">
      <login>test1-laxmi@test.enang.com</login>
      <principal-name>test1 laxmi</principal-name>
      <custom0>408-123-1234</custom0>
    </row>
    <row transcript-id="634" sco-id="126" principal-id="632"
status="completed" score="0" percentage-score="" time-taken="5360"
certificate="634" answered-survey="0">
      <login>testcourse-laxmi@test.enang.com</login>
      <date-taken>2004-07-07T12:37:29.623-07:00</date-taken>
      <principal-name>courseTest1 laxmi</principal-name>
      <custom0 />
    </row>
  </report-quiz-takers>
</results>
```

report-disk-usage

Description

Provides information about how much hard disk space the Breeze content for the current account uses, in bytes. Includes all content and archives.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

`report-disk-usage`

Sample results

```
<results>
  <status code="ok" />
  <report-disk-usage total="849327881" />
</results>
```

report-meeting-attendance

Description

Provides a list of users who have attended the specified meeting. If the meeting hasn't started, the returned data contains no rows.

The data does not include users who were invited but did not attend the meeting.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a meeting.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>transcript-id</code>	Filter and sort
<code>tr.sco-id</code>	Filter and sort
<code>principal-id</code>	Filter and sort
<code>contact-id</code>	Filter and sort
<code>answered-survey</code>	Sort
<code>login</code>	Filter and sort
<code>session-name</code>	Sort
<code>sco-name</code>	Sort
<code>tr.date-created</code>	Filter and sort
<code>tr.date-closed</code>	Filter and sort
<code>participant-name</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`report-meeting-attendance`

Sample results

```
<results>
  <status code="ok" />
  <report-meeting-attendance>
    <row transcript-id="682" sco-id="679" principal-id="8" contact-id="3"
      answered-survey="0">
      <login>lnagaraj@macromedia.com</login>
      <session-name>Enterprise Administrator</session-name>
      <sco-name>test-mtg-2</sco-name>
      <date-created>2004-07-09T14:29:49.217-07:00</date-created>
      <date-closed>2004-07-09T14:33:45.827-07:00</date-closed>
      <participant-name>Enterprise Administrator</participant-name>
    </row>
    <row transcript-id="689" sco-id="679" principal-id="8" contact-id="3"
      answered-survey="0">
      <login>lnagaraj@macromedia.com</login>
      <session-name>Enterprise Administrator</session-name>
      <sco-name>test-mtg-2</sco-name>
      <date-created>2004-07-09T14:35:59.170-07:00</date-created>
      <date-closed>2004-07-09T15:03:05.263-07:00</date-closed>
      <participant-name>Enterprise Administrator</participant-name>
    </row>
  </report-meeting-attendance>
</results>
```

report-meeting-concurrent-users

Description

Indicates the maximum number of users who can participate simultaneously in the specified meeting. This maximum is determined by the account's license.

Parameters

accesskey The access key for your account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

sco-id The ID of a meeting.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-meeting-concurrent-users](#)

Sample results

```
<results>
  <status code="ok" />
  <report-meeting-concurrent-users max-users="426"
    max-participants-freq="1" />
</results>
```

report-meeting-session

Description

Provides information about a specific meeting session, such as the session name, session start and end times, and the number of participants and guests who attended the session.

Each time someone enters an otherwise empty meeting, a new session starts. The session ends when all attendees leave the meeting. If someone subsequently enters the now-empty meeting, a new session starts.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a meeting. This value must match the `sco-id` for a specific meeting session. The list of all sessions and their associated `sco-ids` can be obtained by invoking the [report-meeting-sessions](#) action.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-meeting-session](#)

Sample results

```
<results>
  <status code="ok" />
  <report-meeting-session session-sco-id="671" sco-id="623" num-
    participants="1" num-guests="0">
    <name>test-mtg-1 - 1</name>
    <date-begin>2004-07-09T13:00:32.153-07:00</date-begin>
    <date-end>2004-07-09T13:08:57.483-07:00</date-end>
  </report-meeting-session>
</results>
```

report-meeting-sessions

Description

Provides information about all the sessions of a meeting. Each time someone enters an otherwise empty meeting, a new session starts. The session ends when all attendees leave the meeting. If someone subsequently enters the now-empty meeting, a new session starts.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a meeting.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
sco-id	Filter and sort
num-participants	Sort
num-guests	Sort
num-polls	Sort
name	Filter and sort
date-begin	Filter and sort
date-end	Filter and sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[report-meeting-sessions](#)

Sample results

```
<results>
  <status code="ok" />
  <report-meeting-sessions>
    <row sco-id="622852" num-participants="1" num-guests="0" num-polls="0">
      <name>My Meeting</name>
      <date-begin>2004-03-05T10:33:09.030-08:00</date-begin>
      <date-end>2004-03-05T10:54:09.030-08:00</date-end>
    </row>
    <row sco-id="622852" num-participants="1" num-guests="0" num-polls="0">
      <name>My Meeting</name>
      <date-begin>2004-03-05T11:00:09.030-08:00</date-begin>
    </row>
  </report-meeting-sessions>
</results>
```

report-meeting-session-slots

Description

Provides information about the number of attendees in every ten-minute time slot of the specified meeting session.

Parameters

accesskey The access key for your account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

sco-id The ID of a valid meeting session. You can be obtained this ID by calling the [report-meeting-session-slots](#) action.

Filters

Results cannot be sorted or filtered. The default sort is by the time slot start time.

Returned elements

`report-meeting-session-slots`

Sample results

```
<results>
  <status code="ok" />
  <report-meeting-session-slots>
    <row num-attendees="0">
      <time-slot-begin>2004-02-14T06:30:00.000-08:00</time-slot-begin>
      <time-slot-end>2004-02-14T06:40:00.000-08:00</time-slot-end>
    </row>
    <row num-attendees="0">
      <time-slot-begin>2004-02-14T06:40:00.000-08:00</time-slot-begin>
      <time-slot-end>2004-02-14T06:50:00.000-08:00</time-slot-end>
    </row>
    <row num-attendees="0">
      <time-slot-begin>2004-02-14T06:50:00.000-08:00</time-slot-begin>
      <time-slot-end>2004-02-14T07:00:00.000-08:00</time-slot-end>
    </row>
  </report-meeting-session-slots>
</results>
```

report-meeting-summary

Description

Indicates how many users were invited to the specified meeting and how many invitees and guests attended.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a meeting.

Filters

Results cannot be filtered or sorted.

Returned elements

`report-meeting-summary`

Sample results

```
<results>
  <status code="ok" />
  <report-meeting-summary num-invitees="10" num-invitees-attended="5"
    num-guests-attended="1" />
</results>
```

report-my-courses

Description

Provides information about each course that the logged-in user is enrolled in.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>sco-id</code>	Filter and sort
<code>status</code>	Filter and sort
<code>name</code>	Filter and sort
<code>date-created</code>	Filter and sort
<code>date-modified</code>	Filter and sort
<code>date-begin</code>	Filter and sort
<code>expired</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`my-courses`

Sample results

```
<results>
  <status code="ok" />
  <my-courses>
    <course sco-id="555038" status="active" type="course">
      <name>My Name</name>
      <url>admin.breeze.example.com/p38547860</url>
      <date-created>2004-02-09T14:40:38.497-08:00</date-created>
      <date-modified>2004-02-09T14:40:44.623-08:00</date-modified>
      <date-begin>2004-02-09T14:30:00.000-08:00</date-begin>
      <date-end>2004-02-09T14:30:00.000-08:00</date-end>
      <url-path>/p38547860</url-path>
      <expired>true</expired>
    </course>
    <course sco-id="565003" status="active" type="course">
      <name>quiz 100</name>
      <url>admin.breeze.example.com/p81997938</url>
      <date-created>2004-02-16T21:04:48.750-08:00</date-created>
      <date-modified>2004-02-16T21:33:17.560-08:00</date-modified>
```

```

        <date-begin>2004-02-16T21:00:00.000-08:00</date-begin>
        <url-path>/p81997938</url-path>
        <expired>false</expired>
    </course>
</my-courses>
</results>

```

report-my-meetings

Description

Provides information about each meeting the logged-in user is scheduled to attend.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>sco-id</code>	Sort
<code>status</code>	Sort
<code>permission-id</code>	Sort
<code>active-participants</code>	Sort
<code>name</code>	Sort
<code>date-begin</code>	Sort
<code>date-end</code>	Sort
<code>expired</code>	Sort
<code>duration</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`my-meetings`

Sample results

```

<results>
  <status code="ok" />
  <my-meetings>
    <meeting sco-id="181487" type="meeting" status="active"
      permission-id="view" active-participants="0">
      <name>morris shared</name>
      <domain-name>admin.breeze.example.com</domain-name>
      <url-path>/r99718924</url-path>
    </meeting>
  </my-meetings>
</results>

```

```

    <date-begin>2003-07-14T23:00:00.000-07:00</date-begin>
    <date-end>2003-07-15T09:00:00.000-07:00</date-end>
    <expired>true</expired>
  </meeting>
  <meeting sco-id="505659" type="meeting" status="active"
    permission-id="presenter" active-participants="0">
    <name>new skipper</name>
    <domain-name>admin.breeze.example.com</domain-name>
    <url-path>/p50473388/</url-path>
    <date-begin>2004-01-29T05:30:00.000-08:00</date-begin>
    <date-end>2004-01-29T05:30:00.000-08:00</date-end>
    <expired>true</expired>
  </meeting>
</my-meetings>
</results>

```

report-principal-list

Description

Returns the list of all principals in the account. Similar to the `principal-list` action but returns additional user custom field information. This is the only report where you can filter users on the account custom fields.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>principal-id</code>	Filter and sort
<code>type</code>	Filter and sort
<code>has-children</code>	Filter and sort
<code>is-primary</code>	Filter and sort
<code>name</code>	Filter and sort
<code>login</code>	Filter and sort
<code>all account custom fields</code>	Filter and sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`report-principal-list`

Sample results

```
<results>
  <status code="ok" />
  <report-principal-list>
    <row principal-id="8" type="user" has-children="false" is-
primary="false">
      <name>Enterprise Administrator</name>
      <login>lnagaraj@macromedia.com</login>
      <custom0 />
      <custom1 />
    </row>
    <row principal-id="17" type="admins" has-children="true" is-
primary="true">
      <name>Account Administrators</name>
      <login>Account Administrators</login>
      <custom0 />
      <custom1 />
    </row>
    <row principal-id="123" type="user" has-children="false" is-
primary="false">
      <name>test1 laxmi</name>
      <login>test1-laxmi@test.enang.com</login>
      <custom0>ENGG</custom0>
      <custom1>100-07-7777</custom1>
    </row>
  </report-principal-list>
</results>
```

report-quiz-answer-distribution

Description

Indicates how many users selected a specific answer for all questions on a quiz associated with the specified SCO.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
question-number	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[report-quiz-answer-distribution](#)

Sample results

The following XML contains attributes for answer options 1 through 8, even if the quiz question has fewer possible answers:

```
<results>
  <status code="ok" />
  <report-quiz-answer-distribution>
    <row question-number="1" question-id="257" a1="0" a2="1" a3="0" a4="1"
a5="0" a6="0" a7="0" a8="0" total="2">
      <body>Who is Randy Johnson ?</body>
    </row>
    <row question-number="2" question-id="259" a1="1" a2="0" a3="1" a4="0"
a5="0" a6="0" a7="0" a8="0" total="2">
      <body>Who is Allan Iverson ?</body>
    </row>
  </report-quiz-answer-distribution>
</results>
```

When the SCO doesn't have a quiz, an empty [report-quiz-answer-distribution](#) element is returned, as shown in the following example:

```
<results>
  <status code="ok" />
  <report-quiz-answer-distribution />
</results>
```

report-quiz-definition-answers

Description

Provides information about each of the allowed answers (both correct and incorrect) for the specified quiz. An empty [quiz-definition-answers](#) element is returned when the specified SCO does not have a quiz.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation that contains a quiz.

Filters

Results cannot be filtered or sorted.

The default primary sort is by ascending question ID, and the default secondary sort is by ascending answer ID.

Returned elements

[quiz-definition-answers](#)

Sample results

```
<results>
  <status code="ok" />
  <quiz-definition-answers>
    <row answer-id="1" question-id="257" correct="0">
      <body>22</body>
    </row>
    <row answer-id="2" question-id="257" correct="0">
      <body>0</body>
    </row>
    <row answer-id="3" question-id="257" correct="1">
      <body>4</body>
    </row>
    <row answer-id="1" question-id="258" correct="1">
      <body>150,000,000 kilometers</body>
    </row>
    <row answer-id="2" question-id="258" correct="0">
      <body>1 parsec</body>
    </row>
    <row answer-id="3" question-id="258" correct="0">
      <body>100 kilometers</body>
    </row>
    <row answer-id="4" question-id="258" correct="0">
      <body>300,000 kilometers</body>
    </row>
    <answer-text>ISO-9000</answer-text>
  </row>
  <row answer-id="1" question-id="259" correct="1">
    <body>ECMA-262</body>
  </row>
  <row answer-id="2" question-id="259" correct="0">
    <body>RFC 821</body>
  </row>
  <row answer-id="3" question-id="259" correct="0">
    <body>ECMA-976</body>
  </row>
</quiz-definition-answers>
</results>
```

report-quiz-definition-questions

Description

Provides information about each question in a quiz.

When the specified SCO doesn't contain a quiz, an empty `quiz-definition-questions` element is returned.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
question-number	Filter and sort
question-id	Filter and sort
score	Filter and sort
chapter-id	Filter and sort
body	Filter and sort
date-modified	Filter and sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[quiz-definition-questions](#)

Sample results

```
<results>
  <status code="ok" />
  <quiz-definition-questions>
    <row question-number="1" question-id="257" score="20" chapter-id="1">
      <body>What is 2+2?</body>
      <date-modified>2004-01-13T17:09:40.230-08:00</date-modified>
    </row>
    <row question-number="2" question-id="258" score="20" chapter-id="1">
      <body>How far is it from Earth to the sun?</body>
      <date-modified>2004-01-13T17:09:40.230-08:00</date-modified>
    </row>
    <row question-number="3" question-id="259" score="20" chapter-id="1">
      <body>Which international standard defines ECMAScript?</body>
      <date-modified>2004-01-13T17:09:40.240-08:00</date-modified>
    </row>
  </quiz-definition-questions>
</results>
```

report-quiz-interactions

Description

Provides information about all the interactions that users have had with the specified quiz. An interaction consists of a user making a choice in a quiz, so this report provides information about every answer every user has ever given to questions in the quiz.

In the results for this action, the `score` attribute indicates the score that the quiz-taker received on that question.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>question-number</code>	Filter and sort
<code>name</code>	Sort
<code>question</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[report-quiz-interactions](#)

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-interactions>
    <row question-number="1" transcript-id="730" question-id="257" sco-
id="727" score="0">
      <name>Enterprise Administrator</name>
      <sco-name>quiz-course</sco-name>
      <date-created>2004-07-13T10:54:09.030-07:00</date-created>
      <question>Who is Randy Johnson ?</question>
    </row>
    <row question-number="2" transcript-id="730" question-id="259" sco-
id="727" score="10">
      <name>Enterprise Administrator</name>
      <sco-name>quiz-course</sco-name>
      <date-created>2004-07-13T10:54:09.030-07:00</date-created>
      <question>Who is Allan Iverson ?</question>
    </row>
    <row question-number="1" transcript-id="731" question-id="257" sco-
id="727" score="0">
      <name>Enterprise Administrator</name>
      <sco-name>quiz-course</sco-name>
      <date-created>2004-07-13T10:55:28.763-07:00</date-created>
      <question>Who is Randy Johnson ?</question>
    </row>
    <row question-number="2" transcript-id="731" question-id="259" sco-
id="727" score="0">
      <name>Enterprise Administrator</name>
      <sco-name>quiz-course</sco-name>
```

```

        <date-created>2004-07-13T10:55:28.763-07:00</date-created>
        <question>Who is Allan Iverson ?</question>
    </row>
</report-quiz-interactions>
</results>

```

report-quiz-question-answer-distribution

Description

Indicates how many users selected a particular answer to a specified quiz question.

Use [report-quiz-definition-answers](#) or [report-quiz-definition-questions](#) to determine the ID of a question (to specify as the `question-id` parameter).

Note: Question IDs do not necessarily correspond to question numbers given in a quiz.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`question-id` The ID of a specific question in the quiz.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>answer-id</code>	Filter and sort
<code>correct</code>	Filter and sort
<code>total-responses</code>	Sort
<code>percentage-responded</code>	Sort
<code>answer-text</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[report-quiz-question-answer-distribution](#)

Sample results

```

<results>
  <status code="ok" />
  <report-quiz-question-answer-distribution>
    <row question-id="259" answer-id="1" correct="0" total-responses="1"
      percentage-responded="10">
      <question-text>Which international standard defines
        ECMAScript?</question-text>
    </row>
  </report-quiz-question-answer-distribution>
</results>

```

```

        <answer-text>ISO-9000</answer-text>
    </row>
    <row question-id="259" answer-id="2" correct="0" total-responses="3"
        percentage-responded="30">
        <question-text>Which international standard defines
            ECMAScript?</question-text>
        <answer-text>RFC 821</answer-text>
    </row>
    <row question-id="259" answer-id="3" correct="0" total-responses="2"
        percentage-responded="20">
        <question-text>Which international standard defines
            ECMAScript?</question-text>
        <answer-text>ECMA-976</answer-text>
    </row>
    <row question-id="259" answer-id="4" correct="1" total-responses="4"
        percentage-responded="40">
        <question-text>Which international standard defines
            ECMAScript?</question-text>
        <answer-text>ECMA-262</answer-text>
    </row>
</report-quiz-question-answer-distribution>
</results>

```

report-quiz-question-distribution

Description

Indicates how many users answered each question in the specified quiz correctly.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>question-id</code>	Filter and sort
<code>question-number</code>	Filter and sort
<code>num-correct</code>	Sort
<code>num-incorrect</code>	Sort
<code>percentage-correct</code>	Sort
<code>score</code>	Filter and sort
<code>body</code>	Filter and sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,”](#) on [page 21](#).

Returned elements

[report-quiz-question-distribution](#)

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-question-distribution>
    <row question-id="257" question-number="1" num-correct="8"
      num-incorrect="0" percentage-correct="100" score="20">
      <body>What is 2+2?</body>
    </row>
    <row question-id="258" question-number="2" num-correct="8"
      num-incorrect="0" percentage-correct="100" score="20">
      <body>How far is it from Earth to the sun?</body>
    </row>
    <row question-id="259" question-number="3" num-correct="4"
      num-incorrect="6" percentage-correct="40" score="20">
      <body>Which international standard defines ECMAScript?</body>
    </row>
  </report-quiz-question-distribution>
</results>
```

report-quiz-question-response

Description

Provides a list of all answers that users have given to a particular quiz question.

Use [report-quiz-definition-answers](#) or [report-quiz-definition-questions](#) to determine the ID of a question (to use as the `question-id` parameter).

Note: Question IDs do not necessarily correspond to question numbers indicated in the quiz.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`question-id` The ID of a specific question in the quiz.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>principal-id</code>	Sort
<code>answer-text</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,”](#) on [page 21](#).

Returned elements

[report-quiz-question-response](#)

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-question-response>
    <row principal-id="506299" question-id="257" answer-id="2">
      <body>What is 2+2?</body>
      <answer-text>5</answer-text>
    </row>
    <row principal-id="557527" question-id="257" answer-id="2">
      <body>What is 2+2?</body>
      <answer-text>5</answer-text>
    </row>
    <row principal-id="181255" question-id="257" answer-id="3">
      <body>What is 2+2?</body>
      <answer-text>4</answer-text>
    </row>
  </report-quiz-question-response>
</results>
```

report-quiz-question-totals

Description

Indicates, for the specified quiz question, the total number of users who answered the question and the number of users who answered the question correctly.

Use [report-quiz-definition-answers](#) or [report-quiz-definition-questions](#) to determine the ID of a question (to use as the `question-id` parameter).

Note: Question IDs do not necessarily correspond to question numbers indicated in the quiz.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`question-id` The ID of a specific question in the quiz.

`sco-id` The ID of a presentation that contains a quiz.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-quiz-question-totals](#)

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-question-totals question-id="260" total-responses="8"
    num-correct-answers="5" />
</results>
```

report-quiz-summary

Description

Provides information about the results of a quiz.

Parameters

accesskey The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

sco-id The ID of a presentation that contains a quiz.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-quiz-results](#), [report-quiz-summary](#)

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-results average-score="95" low-score="90" high-score="100"
    numtaken="8" numdistincttaken="1" />
  <report-quiz-summary sco-id="217706" numquestions="5" maxpossiblescore="100"
    passingscore="50" />
</results>
```

report-quiz-takers

Description

Provides information about everyone who has taken a particular quiz.

To get a listing of SCOs that a specified principal viewed, pass the `principal-id` parameter instead of the `sco-id` parameter.

In addition, the list of the SCOs can be restricted to a specific type by passing an optional parameter `type`.

Type	Description
type=course	Lists all courses the user took.
type=presentation	Lists all content the user viewed.
type= meeting	Lists all meetings the user attended.

If a type parameter is not specified, the list of all courses, content, and meetings attended by the user is returned.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation that contains a quiz.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>transcript-id</code>	Filter and sort
<code>principal-id</code>	Filter and sort
<code>contact-id</code>	Filter and sort
<code>status</code>	Sort
<code>score</code>	Filter and sort
<code>percentage-score</code>	Sort
<code>time-taken</code>	Sort
<code>answered-survey</code>	Sort
<code>login</code>	Filter and sort
<code>date-taken</code>	Sort
<code>principal-name</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`report-quiz-takers`

Sample results

```
<results>
  <status code="ok" />
  <report-quiz-takers>
    <row transcript-id="730" sco-id="727" principal-id="8" contact-id="3"
      status="user-passed" score="10" percentage-score="50" time-taken="70076"
      certificate="730" answered-survey="0">
      <name>quiz-course</name>
      <login>lnagaraj@macromedia.com</login>
      <date-taken>2004-07-13T10:54:09.030-07:00</date-taken>
      <principal-name>Enterprise Administrator</principal-name>
    </row>
```

```

    <row transcript-id="731" sco-id="727" principal-id="8" contact-id="3"
    status="user-failed" score="0" percentage-score="0" time-taken="4656"
    certificate="" answered-survey="0">
      <name>quiz-course</name>
      <login>lnagaraj@macromedia.com</login>
      <date-taken>2004-07-13T10:55:28.763-07:00</date-taken>
      <principal-name>Enterprise Administrator</principal-name>
    </row>
  </report-quiz-takers>
</results>

```

Caution: There was an issue with the formatting of the `time-taken` XML tag, which has been fixed in the Breeze 4.1 updater. If you see incorrect formatting in the content of a `time-taken` XML tag, download the updater from the [Licensed Support Center](#).

report-quotas

Description

Returns information on the account quotas.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-quotas](#)

Sample results

```

<results>
  <status code="ok" />
  <report-quotas>
    <quota acl-id="7" quota-id="2" used="0" limit="unlimited" soft-
    limit="1000000000">
      <date-begin>1999-12-31T16:00:00.000-08:00</date-begin>
      <date-end>2999-12-31T16:00:00.000-08:00</date-end>
    </quota>
    <quota acl-id="7" quota-id="download-quota" used="0" limit="50000"
    soft-limit="50000">
      <date-begin>1999-12-31T16:00:00.000-08:00</date-begin>
      <date-end>2999-12-31T16:00:00.000-08:00</date-end>
    </quota>
  </report-quotas>
</results>

```

report-sco-slides

Description

Indicates how many times, and how recently, each slide in a presentation has been viewed.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
slide	Sort
views	Sort
last-viewed	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`report-sco-slides`

Sample results

```
<results>
  <status code="ok" />
  <report-sco-slides>
    <row slide="1" views="5">
      <last-viewed>2004-01-13T16:35:39.517-08:00</last-viewed>
    </row>
    <row slide="2" views="8">
      <last-viewed>2004-01-13T16:35:42.350-08:00</last-viewed>
    </row>
    <row slide="3" views="7">
      <last-viewed>2004-01-13T16:35:44.933-08:00</last-viewed>
    </row>
    <row slide="4" views="8">
      <last-viewed>2004-01-13T16:35:47.640-08:00</last-viewed>
    </row>
    <row slide="5" views="8">
      <last-viewed>2004-01-13T16:35:51.723-08:00</last-viewed>
    </row>
  </report-sco-slides>
</results>
```

report-sco-views

Description

Indicates how many times, and how recently, the specified presentation has been viewed.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a presentation.

Filters

Results cannot be filtered or sorted.

Returned elements

`report-sco-views`

Sample results

```
<results>
  <status code="ok" />
  <report-sco-views>
    <row sco-id="217706" type="presentation" is-folder="0" views="24">
      <name>Quiz test 1</name>
      <last-viewed-date>2004-01-13T17:11:47.923-08:00</last-viewed-date>
    </row>
  </report-sco-views>
</results>
```

report-survey-question-response

Description

Returns the list of all users who answered the specified question and their answer.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`question-id` The ID of the question.

`sco-id` The ID of the SCO.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
principal-id	Sort
user-name	Sort
response	Filter and sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

[report-survey-question-response](#)

Sample results

```
<results>
  <status code="ok" />
  <report-survey-question-response>
    <row principal-id="8" question-number="1">
      <user-name>Enterprise Administrator</user-name>
      <question>how are you?</question>
      <response>fine</response>
    </row>
    <row principal-id="123" question-number="1">
      <user-name>test1 laxmi</user-name>
      <question>how are you?</question>
      <response>ok</response>
    </row>
  </report-survey-question-response>
</results>
```

report-training-concurrent-users

Description

Indicates the maximum number of concurrent course users for the specified account.

Parameters

accesskey The access key for your account.

Note: The **accesskey** parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[report-training-concurrent-users](#)

Sample results

```
<results>
  <status code="ok" />
  <report-training-concurrent-users account-id="181224" max-num-sessions="2"
    limit="1000000000">
    <name>Professor Test Account</name>
  </report-training-concurrent-users>
</results>
```

sco-build

Description

Causes the Breeze server to build the specified SCO when you create presentations with the XML API. (For more information, see [sco-update](#) and [sco-upload](#)). No one can view a SCO until it has been built. Call this action after you upload a SCO using the [sco-upload](#) action.

If you upload a new version of an existing SCO, you must rebuild the SCO using the `sco-build` action; otherwise, a user viewing the SCO continues to see the old version.

You can call the [sco-info](#) action to obtain the current build status of the SCO. The SCO status is initially be *in-progress*, which means the content is being built. When the status becomes *active*, the content has finished building and is accessible.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a SCO to build.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

See also

[sco-upload](#)

sco-contents

Description

Provides a list of the SCOs in a specified folder.

Because folders are SCOs, the returned list includes the folders that are contained in the specified folder. The returned list does not include the contents of those subfolders.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a folder.

Filters

The following table lists the fields that the data can be filtered and sorted on:

Field	Filter/Sort
<code>sco-id</code>	Filter and sort
<code>status</code>	Filter and sort
<code>type</code>	Filter and sort
<code>is-folder</code>	Sort
<code>name</code>	Filter and sort
<code>date-begin</code>	Filter and sort
<code>date-end</code>	Filter and sort
<code>date-modified</code>	Filter and sort
<code>duration</code>	Sort

For more information about filtering and sorting, see [Chapter 2, “Working with Filters,” on page 21](#).

Returned elements

`SCOS`

Sample results

The `name` and `date-modified` elements are always returned, but the `date-begin`, `date-end` and `duration` elements are present only for SCOs whose `type` attribute is `course` or `meeting`.

```
<results>
  <status code="ok" />
  <sco>
    <sco sco-id="505677" status="no-content" type="folder" is-folder="1">
      <name>PPT meeting tests</name>
      <date-modified>2004-01-29T06:22:49.157-08:00</date-modified>
    </sco>
  </sco>
```

```

<sco sco-id="193496" status="active" type="meeting" is-folder="0">
  <name>My Name</name>
  <date-begin>2003-08-27T00:00:00.000-07:00</date-begin>
  <date-end>2003-08-27T01:00:00.000-07:00</date-end>
  <date-modified>2003-08-26T23:33:20.300-07:00</date-modified>
  <duration>1900-01-02T/+:00:00.000+00:00</duration>
</sco>
<sco sco-id="505323" status="active" type="meeting" is-folder="0">
  <name>My Name 2</name>
  <date-begin>2004-01-29T01:30:00.000-08:00</date-begin>
  <date-end>2004-01-29T01:30:00.000-08:00</date-end>
  <date-modified>2004-01-29T01:35:26.433-08:00</date-modified>
  <duration>1900-01-02T/*:00:00.000+00:00</duration>
</sco>
</scos>
</results>

```

sco-delete

Description

Deletes one or more SCOs. If the specified `sco-id` is a folder, all the contents of the specified folder are also deleted.

To delete multiple SCOs, specify multiple `sco-id` parameters. For more information, see [“About parameters” on page 15](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a SCO.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```

<results>
  <status code="ok" />
</results>

```

sco-info

Description

Provides information about a SCO. For information about SCOs, see [“About principals, SCOs, and IDs” on page 16](#).

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a SCO.

Filters

Results cannot be filtered or sorted.

Returned elements

The returned tags depend on the specified item's SCO type.

Some tags that might be returned include the following:

[sco](#), [sco-author](#), [sco-data](#), [source-sco](#)

Sample results

The following is sample XML from a SCO whose `type` attribute is `sample`:

```
<results>
  <status code="ok" />
  <sco folder-id="15" host-id="" lang="en" learning-time="" sco-data-id="625"
sco-id="623" source-sco-id="" status="active" tree-id="14" type="meeting">
    <date-begin>2004-07-07T10:00:00.000-07:00</date-begin>
    <date-created>2004-07-07T10:05:33.200-07:00</date-created>
    <date-end>2004-07-07T11:00:00.000-07:00</date-end>
    <date-modified>2004-07-09T15:09:41.390-07:00</date-modified>
    <name>test-mtg-1</name>
    <url-path>/r66599952</url-path>
  </sco>
  <sco-data byte-count="" height="" length="" sco-data-id="625" slide-count=""
version="0" width="">
    <date-created>2004-07-07T10:05:33.200-07:00</date-created>
    <date-modified>2004-07-07T10:05:33.200-07:00</date-modified>
    <physical-path>7/623</physical-path>
  </sco-data>
  <source-sco />
</results>
```

The following is sample XML from a SCO whose `type` attribute is `presentation`:

```
<results>
  <status code="ok" />
  <sco folder-id="24" host-id="" lang="en" learning-time="" sco-data-id="29"
sco-id="27" source-sco-id="" status="active" tree-id="11"
type="presentation">
```

```

    <author-info-1>l</author-info-1>
    <author-info-2>n</author-info-2>
    <date-created>2004-06-30T11:45:03.717-07:00</date-created>
    <date-modified>2004-06-30T11:53:33.250-07:00</date-modified>
    <name>test-slide</name>
    <url-path>p27916084</url-path>
  </sco>
  <sco-data byte-count="766632" height="540" length="11280" sco-data-id="29"
  slide-count="2" version="0" width="720">
    <date-created>2004-06-30T11:45:03.717-07:00</date-created>
    <date-modified>2004-06-30T11:53:33.090-07:00</date-modified>
    <physical-path>7/27</physical-path>
  </sco-data>
  <source-sco />
</results>

```

The following is sample XML from a SCO whose type attribute is folder:

```

<results>
  <status code="ok" />
  <sco folder-id="24" host-id="" lang="en" learning-time="" sco-data-id="746"
  sco-id="744" source-sco-id="" status="active" tree-id="11" type="folder">
    <date-created>2004-07-14T12:19:23.340-07:00</date-created>
    <date-modified>2004-07-14T12:19:23.340-07:00</date-modified>
    <name>test2</name>
    <url-path>p33526645</url-path>
  </sco>
  <sco-data byte-count="" height="" length="" sco-data-id="746" slide-count=""
  version="0" width="">
    <date-created>2004-07-14T12:19:23.340-07:00</date-created>
    <date-modified>2004-07-14T12:19:23.340-07:00</date-modified>
    <physical-path>7/744</physical-path>
  </sco-data>
  <source-sco />
</results>

```

The following is sample XML from a SCO whose type attribute is course:

```

<results>
  <status code="ok" />
  <sco folder-id="10" host-id="" lang="en" learning-time="" sco-data-id="722"
  sco-id="727" source-sco-id="720" status="active" tree-id="10" type="course">
    <date-begin>2004-07-13T10:30:00.000-07:00</date-begin>
    <date-created>2004-07-13T10:53:47.450-07:00</date-created>
    <date-modified>2004-07-13T10:53:51.763-07:00</date-modified>
    <name>quiz-course</name>
    <url-path>p38917184</url-path>
  </sco>
  <sco-data byte-count="1400615" height="540" length="31200" sco-data-id="722"
  slide-count="3" version="0" width="720">
    <date-created>2004-07-13T10:46:48.000-07:00</date-created>
    <date-modified>2004-07-13T10:47:23.653-07:00</date-modified>
    <physical-path>7/720</physical-path>
  </sco-data>
  <source-sco>

```

```

    <source-sco folder-id="12" host-id="" lang="en" learning-time="" sco-data-
id="722" sco-id="720" source-sco-id="" status="active" tree-id="11"
type="presentation">
    <author-info-1>1</author-info-1>
    <date-created>2004-07-13T10:46:48.000-07:00</date-created>
    <date-modified>2004-07-13T10:47:23.810-07:00</date-modified>
    <name>test-quiz</name>
    <url-path>/p67601053</url-path>
  </source-sco>
</source-sco>
</results>

```

sco-move

Description

Moves a SCO from one folder to another.

For information about rules regarding moves and their necessary permissions, see Breeze Help.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`folder-id` The ID of the destination folder.

`sco-id` The ID of the SCO to move.

Filters

Results cannot be filtered or sorted.

Returned elements

`status`

Sample results

```

<results>
  <status code="ok" />
</results>

```

sco-nav

Description

Describes the folder hierarchy that contains the specified SCO.

In the returned results, the `sco` tag's `depth` attribute indicates how many hierarchical levels away the folder is from the specified SCO. That is, a `depth` attribute equal to 0 indicates the SCO itself; a `depth` of 1 indicates the folder that contains the SCO; and so on.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`sco-id` The ID of a SCO.

Filters

Results cannot be filtered or sorted.

Returned elements

`sco-nav`

Sample results

```
<results>
  <status code="ok" />
  <sco-nav>
    <sco sco-id="181230" tree-id="181230" type="folder" depth="2">
      <name>User Meetings</name>
    </sco>
    <sco sco-id="181576" tree-id="181230" type="folder" depth="1">
      <name>tenant@example.com</name>
    </sco>
    <sco sco-id="503545" tree-id="181230" type="meeting" depth="0">
      <name>created with 402</name>
    </sco>
  </sco-nav>
</results>
```

sco-search

Description

Provides a list of all SCOs that match the search text.

For every type of SCO, the title and summary are searched. For presentations and other SCOs that contain text, the contents of the SCO are also searched.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`query` A string to search for. You can use the backslash (\) character before the following characters to escape them in the query string:

+ - && || ! () { } [] ^ " ~ * ? : \

Filters

You can restrict the returned data to a specific number of rows using the `filter-rows` and `filter-start` parameters. For example, the following code returns the first five rows of data:

`filter-rows=5&filter-start=0`

Note: The `filter-rows` parameter indicates the number of rows to be returned, and `filter-start` indicates where the row count should begin. For more information about filters, see [Chapter 2, “Working with Filters,”](#) on page 21.

You can also use the `filter-type` parameter to restrict the returned data based on the type attribute of the SCO. For example, the following returns data of type presentation only:

```
filter-type=0
```

The following returns data of type archives only:

```
filter-type=8
```

Note: If the parameter is not specified, all matches are returned.

Returned elements

`sco-search-info`

Sample results

```
<results>
  <status code="ok" />
  <sco-search-info>
    <sco sco-id="202014" folder-id="202008" type="presentation"
      slide-count="4" length="20000">
      <name>Quiz 1</name>
      <url-path>/p92628681/</url-path>
      <date-created>2003-11-26T22:52:28.973-08:00</date-created>
      <date-modified>2003-11-26T22:52:37.867-08:00</date-modified>
      <author-info-1>lnagaraj@macromedia.com</author-info-1>
      <hit>1</hit>
      <hit>4</hit>
    </sco>
    <sco sco-id="202015" folder-id="202008" type="presentation"
      slide-count="7" length="35000">
      <name>More broken quizzes</name>
      <url-path>/p49954414/</url-path>
      <date-created>2003-11-26T23:02:52.870-08:00</date-created>
      <date-modified>2003-11-26T23:03:23.313-08:00</date-modified>
      <author-info-1>foo-account@macromedia.com</author-info-1>
      <hit>1</hit>
      <hit>4</hit>
      <hit>5</hit>
      <hit>7</hit>
    </sco>
  </sco-search-info>
</results>
```

sco-shortcuts

Description

Provides IDs for a set of folders containing content relevant to the logged-in user: a folder for the user to place meetings, a folder for the user to place content, and so on.

To determine the URL of a SCO, concatenate the domain name returned by this action with the URL returned by the `sco-info` action in the `url-path` tag.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

Filters

Results cannot be filtered or sorted.

Returned elements

[shortcuts](#)

Sample results

```
<results>
  <status code="ok" />
  <shortcuts>
    <sco tree-id="181230" sco-id="181576" type="my-meetings">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
    <sco tree-id="181227" sco-id="181412" type="my-content">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
    <sco tree-id="181225" sco-id="181225" type="content">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
    <sco tree-id="181226" sco-id="181226" type="courses">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
    <sco tree-id="181227" sco-id="181227" type="user-content">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
    <sco tree-id="181228" sco-id="181228" type="meetings">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
    <sco tree-id="181230" sco-id="181230" type="user-meetings">
      <domain-name>http://admin.breeze.example.com</domain-name>
    </sco>
  </shortcuts>
</results>
```

See also

[sco-info](#)

sco-update

Description

Creates or updates presentations, courses, and meetings.

If you provide a `sco-id` parameter, this action updates the specified SCO.

If you provide a `folder-id` parameter, this action creates a new SCO. If you create a new SCO, the returned XML provides information about the new SCO, including the SCO's ID.

You must provide only one of the following two parameters: `folder-id` or `sco-id`.

After you call this action, you must call the [permissions-update](#) action to assign permissions. For information about permissions, see [“About security” on page 17](#).

For more information about SCOs, see [“About principals, SCOs, and IDs” on page 16](#).

The following examples show uses for the `sco-update` action.

To use `sco-update` with a Breeze presentation:

1. Call `sco-update` with the `folder-id` parameter to create a SCO.

Pass the following parameters: `accesskey`, `folder-id`, `description`, `name`, `author-info-1` (the name of the author), `author-info-2` (the author's title), `author-info-3` (the company name), `type=presentation`, and `lang`.

2. Call [sco-upload](#) to associate content with this SCO.

Note: See [sco-upload](#) for a sample of an HTML form that associates content with a presentation created through the XML API.

3. Call [sco-build](#) to build the SCO.
4. Call the [permissions-update](#) action to assign permissions to the SCO.

To use `sco-update` with a Breeze course:

1. Call `sco-update` to create a course and associate pre-existing content with the course through the `source-sco-id` parameter.

Pass the following parameters: `accesskey`, `folder-id`, `date-begin`, `date-end`, `description`, `name`, `type=course`, `source-sco-id`, and `sco-tag`.

2. Call the [permissions-update](#) action to add enrollees to the course.

To use `sco-update` with a Breeze meeting:

1. Call `sco-update` to create the meeting.

Pass the following parameters: `accesskey`, `folder-id`, `date-begin`, `date-end`, `description`, `name`, and `type=meeting`.

Note: You must use the Breeze Manager to associate content with a meeting that was created using the XML API.

2. Call [permissions-update](#) with the following parameter values to set public or private access to the meeting:

```
// private:  
principal-id=public-access&permission-id=denied  
// public:  
principal-id=public-access&permission-id=view-hidden
```

3. Call the [permissions-update](#) action to assign presenter or participant permissions to the meeting.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`date-begin` The scheduled beginning date and time. The date format is ISO 8601. (Used only for meetings and courses.)

`date-end` The scheduled ending date and time. The date format is ISO 8601. (Used only for meetings and courses.)

`description` A description of the SCO to be read by users.

`email` The e-mail address of the contact person for the SCO. (Used only for presentations.)

`first-name` The first name of the contact person for the SCO. (Used only for presentations.)

`folder-id` The ID of the folder in which the server will create the new SCO. This parameter is optional and cannot be specified with `sco-id`.

`last-name` The last name of the contact person for the SCO. (Used only for presentations.)

`name` The name of the SCO.

`sco-id` The ID of a SCO to update. This parameter is optional and cannot be specified with `folder-id`.

`sco-tag` A label for the SCO. Can contain any arbitrary data; for example, for an Economics 101 course, you might specify 101 for the `sco-tag` parameter. (Used only for courses.)

`source-sco-id` The ID of the SCO's content. (Used only for courses.)

`type` The type of the new SCO. For information about types, see [“About principals, SCOs, and IDs” on page 16](#).

Filters

Results cannot be filtered or sorted.

Returned elements

If you update an existing SCO, no elements are returned.

If you create a new SCO, `sco`.

Sample results

The following sample XML is returned when a new SCO is created:

```
<results>
  <status code="ok" />
  <sco folder-id="181412" type="folder" sco-id="2501961" status="active"
    author-contact-id="2501962" tree-id="181227" sco-data-id="2501963">
    <url-path>/p54135747/</url-path>
    <name>Test SCO</name>
  </sco>
</results>
```

The following sample XML is returned when a presentation SCO is created:

```
<results>
<status code="ok" />
  <sco folder-id="945" type="presentation" lang="en" sco-id="2152" status="no-
content" tree-id="11" sco-data-id="2154">
    <author-info-2>sw</author-info-2>
    <author-info-3>macr</author-info-3>
    <url-path>/p74161709/</url-path>
    <description>testXMLAPIpreso-New</description>
    <author-info-1>laxmi n</author-info-1>
    <name>testAPI-presno-New</name>
  </sco>
</results>
```

To update the title, summary, speaker name, title, language, and company, replace the parameter `folder-id` with the parameter `sco-id`.

To upload a different SCO and rebuild the presentation, call `sco-upload` and `sco-build` in that order.

The following sample results are returned when a presentation is updated:

```
<results>
  <status code="ok" />
</results>
```

The following sample XML is returned when a course is created:

```
<results>
<status code="ok" />
  <sco folder-id="954" lang="1" type="course" sco-id="1520" source-sco-
id="720" status="active" tree-id="10" sco-data-id="722">
    <date-begin>2004-07-26T24:41:58.330-07:00</date-begin>
    <date-end>2004-07-28T21:41:58.330-07:00</date-end>
    <sco-tag>test-3</sco-tag>
    <url-path>/p41607453/</url-path>
    <description>testXMLAPICourse-3</description>
    <name>testAPI-course-3</name>
  </sco>
</results>
```

To update the start date, end date, course ID, summary, description, name, and content associated with a course, replace the parameter `folder-id` with the parameter `sco-id`.

The following sample XML is returned when updating a course:

```
<results>
  <status code="ok" />
</results>
```

The following sample XML is returned when creating a meeting:

```
<results>
<status code="ok" />
  <sco folder-id="1426" lang="en" type="meeting" sco-id="1429" status="active"
tree-id="14" sco-data-id="1431">
    <date-begin>2004-07-22T21:41:58.330-07:00</date-begin>
```

```

    <date-end>2004-07-22T21:51:58.330-07:00</date-end>
    <url-path>/r54578750/</url-path>
    <description>testXMLAPI-mtg</description>
    <name>testAPI-mtg</name>
  </sco>
</results>

```

You can update meeting start and end times, name, and description by replacing the parameter `folder-id` with the parameter `sco-id`.

The following sample XML is returned when updating a meeting:

```

<results>
  <status code="ok" />
</results>

```

sco-upload

Description

Uploads a file to the Breeze server when creating a presentation using the XML API. (For more information, see [sco-update](#) and [sco-build](#)).

You must send the parameters for this action using the multipart/form-data media type. The easiest way to create the `file` parameter is to use a form control of type "file" in a form on a web page.

After you upload a file, call the [sco-build](#) action to build it.

The following sample HTML is a form for associating content with a presentation created with the XML API:

```

<FORM action="http://<domain-name>/api/xml?accesskey=xxxxxxxxxxxx&action=sco-upload&sco-id=xx&summary=xxx&title=xxx" enctype="multipart/form-data" method="post">
  <P>
    What files are you sending?
    <INPUT type="file" name="file">
    <BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset">
</FORM>

```

This form uploads a single file. If you need to upload multiple files, for example, a PPT and a PPC file, you must pass an additional `file` parameter:

```

<FORM action="http://<domain-name>/api/xml?accesskey=xxxxxxxxxxxx&action=sco-upload&sco-id=xxx&summary=xxx&title=xxx" enctype="multipart/form-data" method="post">
  <P>
    PPT files you are sending <INPUT type="file" name="file"><BR>
    PPC files you are sending <INPUT type="file" name="file"><BR>

    <INPUT type="submit" value="Send"> <INPUT type="reset">
</FORM>

```

Note: Call the action [sco-info](#) to obtain the latest SCO status. The sco status will initially be in-progress, which means the content is being built. When the status becomes active, the content has finished building and is now accessible.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`file` The file that you want to upload. This should be the actual data for the file, not just a filename or path.

`sco-id` The ID of the new SCO. You must have already set a type for the SCO; for example, by using [sco-update](#).

`summary` A brief summary of general information about the SCO. The summary appears in content listings and is returned by the [sco-info](#) action. This parameter is optional.

`title` The title of the new SCO.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```

See also

[sco-build](#)

trees-info

Description

Returns the list of trees that belong to the specified account.

Parameters

`accesskey` The access key for the account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`account-id` The ID of the account.

Filters

Results cannot be filtered or sorted. The default sort is by ascending `tree-id`.

Returned elements

trees

Sample results

```
<results>
  <status code="ok" />
  <trees>
    <tree account-id="7" tree-id="9" sco-id="9" type="content">
      <domain-name>nagarajan.macromedia.com</domain-name>
    </tree>
    <tree account-id="7" tree-id="10" sco-id="10" type="courses">
      <domain-name>nagarajan.macromedia.com</domain-name>
    </tree>
    <tree account-id="7" tree-id="11" sco-id="11" type="user-content">
      <domain-name>nagarajan.macromedia.com</domain-name>
    </tree>
    <tree account-id="7" tree-id="13" sco-id="13" type="meetings">
      <domain-name>nagarajan.macromedia.com</domain-name>
    </tree>
    <tree account-id="7" tree-id="14" sco-id="14" type="user-meetings">
      <domain-name>nagarajan.macromedia.com</domain-name>
    </tree>
  </trees>
</results>
```

user-accounts

Description

Provides a list of the accounts to which the specified user belongs.

In an enterprise (nonhosted) Breeze system, there is usually only one account; if you have an enterprise system, you are unlikely to need this action.

Parameters

accesskey The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

login The user's login name, which is usually the user's e-mail address.

password The user's password.

Filters

Results cannot be filtered or sorted. Default sorting is by the `account-name`.

Returned elements

users

Sample results

```
<results>
<status code="ok" />
<users>
  <user user-id="211729" account-id="211655">
    <name>james@example.com</name>
    <date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
  </user>
  <user user-id="212988" account-id="212928">
    <name>smith@example.com</name>
    <date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
  </user>
  <user user-id="214251" account-id="214191">
    <name>jones@example.com</name>
    <date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
  </user>
</users>
</results>
```

user-update-pwd

Description

Changes a user's password.

Parameters

`accesskey` The access key for your account.

Note: The `accesskey` parameter is required only in Breeze 4.0; it is not required for Breeze 4.1. If you include this parameter in Breeze 4.1, it is ignored.

`password` The new password.

`password-old` The user's current password.

`password-verify` A second copy of the new password, for verification.

`user-id` The ID of the user.

Filters

Results cannot be filtered or sorted.

Returned elements

None.

Sample results

```
<results>
  <status code="ok" />
</results>
```


CHAPTER 5

XML Results Reference

This chapter describes the XML tags, data, and status codes returned by the Macromedia Breeze server in response to an action. For information about the actions you can call on the server, see [Chapter 4, “Action Reference,” on page 37](#).

About returned XML code

When you call an action on the Breeze server, the server returns a block of XML code.

The parent tag of the XML code is always a `results` tag. The `results` tag contains a `status` tag, which indicates whether the action was completed successfully. In most cases, if the action was completed successfully, the `results` tag also contains other tags that contain the data returned by the action.

For example, a `common-info` action might return the following results:

```
<results>
  <status code="ok" />
  <common locale="en">
    <cookie>breezm5qtgnye46zpckbf.MARIANNE</cookie>
    <date>2004-02-12T15:53:19.797-07:00</date>
    <host>http://admin.breeze.example.com</host>
    <url>/api/xml?accesskey=yyyyyyyyyyyyyyyy&action=common-info</url>
    <version>breeze_402_r116</version>
    <account account-id="222914" />
    <user user-id="503562">
      <login>gerweck@example.com</login>
    </user>
  </common>
</results>
```

This chapter organizes the tags alphabetically; each tag has its own section.

For information on possible status values, see [status](#).

For information about the tags returned for a particular action, see the documentation for that action in [Chapter 4, “Action Reference,” on page 37](#).

If there is no data available for a tag that would normally be returned in response to an action, the tag isn't returned. In the previous example, if the calling application didn't specify a logged-in user, there would be no `user` tag in the returned results.

When your web application receives the returned XML code, it must parse the code to determine what the returned values are. For examples, see [Chapter 3, “Common Tasks,”](#) on page 27.

Alphabetical list of XML tags

The following table is an alphabetical list of the XML tags documented in this reference:

Tag	Description
<code>access-key (container)</code>	Contains information about a special access key provided by a SCO.
<code>access-key (key)</code>	Contains a special access key for use in self-registration.
<code>access-key-group</code>	Contains information about which group the user automatically joins by self-registering using the special access key.
<code>access-keys</code>	Contains one or more access keys.
<code>account</code>	The account ID.
<code>action</code>	The name of an action.
<code>answer-correct</code>	The text for a correct answer.
<code>answer-text</code>	The text of an answer to a quiz question as part of the results returned by a report action.
<code>body</code>	The text of a returned item, such as the text of a quiz question in a quiz-related report.
<code>comment</code>	Any comments on the custom field (may be 0 to 60 characters long).
<code>common</code>	Contains information about the server and the logged-in user.
<code>contact</code>	Contains information about a contact.
<code>cookie</code>	The value of the Java servlet session cookie, named BREEZESession.
<code>course</code>	Contains information about a course.
<code>custom-fields</code>	Contains information about the custom fields of an account.
<code>date</code>	The current date and time on the server, in ISO 8601 date format.
<code>date-begin</code>	The date and time that a SCO is, or was, scheduled to begin, in ISO 8601 date format.
<code>date-closed</code>	The time at which the user closed the window that contains the meeting, thereby leaving the meeting, in ISO 8601 date format.

Tag	Description
<code>date-created</code>	The date and time that a SCO was created, in ISO 8601 date format.
<code>date-end</code>	The date and time that a SCO is, or was, scheduled to end, in ISO 8601 date format.
<code>date-expired</code>	The date that an account is, or was, scheduled to expire, in ISO 8601 date format.
<code>date-last-taken</code>	The date that a course was last taken, in ISO 8601 date format.
<code>date-modified</code>	The date that something was most recently modified, in ISO 8601 date format.
<code>date-taken</code>	The date that a particular user took a particular quiz, in ISO 8601 date format.
<code>date-time-attempted</code>	The date that a particular user took a particular quiz.
<code>description</code>	A description of a principal.
<code>domain-name</code>	The fully qualified domain name of the location of a SCO.
<code>duration</code>	The duration of a SCO.
<code>email</code>	A user's e-mail address.
<code>expired</code>	A Boolean value indicating whether a SCO has expired.
<code>field</code>	Contains information about custom fields.
<code>first-name</code>	A user's first name.
<code>hit</code>	A number indicating which slide contains a search term.
<code>host</code>	The URL of the host that the Breeze server is running on.
<code>invalid</code>	Contains information about why a particular call to an action is invalid.
<code>last-name</code>	A user's last name.
<code>last-viewed</code>	The date and time of the most recent time that a SCO was viewed, in ISO 8601 date format.
<code>last-viewed-date</code>	The most recent date and time that the SCO was viewed, in ISO 8601 date format.
<code>login</code>	A user's login name, which is usually the user's e-mail address.
<code>meeting</code>	Contains information about a meeting.
<code>my-courses</code>	Contains information about one or more courses.
<code>my-meetings</code>	Contains information about one or more meetings.
<code>name</code>	A user's name.
<code>participant-name</code>	The name of a participant in a meeting.

Tag	Description
<code>permission</code>	Contains information about a specific principal and its permissions.
<code>permissions</code>	Contains information about one or more principals and their permissions.
<code>physical-path</code>	The path on the server to a SCO.
<code>preferences</code>	Contains information about user preferences.
<code>presentation-name</code>	The name of a Breeze presentation.
<code>principal</code>	Contains information about a principal.
<code>principal-custom-field-values</code>	Contains information about the values of a custom field in a principal.
<code>principal-list</code>	Contains one or more <code>principal</code> tags.
<code>principal-name</code>	The name of a principal.
<code>question</code>	The text of a quiz question.
<code>question-text</code>	The text of a quiz question.
<code>quiz-definition-answers</code>	Contains one or more <code>row</code> tags, each containing information about a particular answer to a particular quiz question.
<code>quiz-definition-questions</code>	Contains one or more <code>row</code> tags, each containing information about a particular quiz question.
<code>quiz-name</code>	The name of a quiz.
<code>quota</code>	Contains information on the account quotas.
<code>report-account-meeting-attendance</code>	Contains <code>row</code> tags that contain information about the meeting attendance log for the account.
<code>report-active-meeting-presenters</code>	Contains a list of the users who are currently presenting meetings.
<code>report-active-meetings</code>	Contains a list of meetings that are currently in progress.
<code>report-bandwidth</code>	The total bandwidth consumed in the current account since the account was created, in bytes.
<code>report-bulk-content-quiz</code>	Contains rows that contain information about all content quizzes, including a list of quizzes.
<code>report-bulk-content-quiz-results</code>	Contains row information about the results for a content quiz.
<code>report-bulk-course-quiz</code>	Contains rows that contain information about all course quizzes, including a list of quizzes.
<code>report-bulk-course-quiz-results</code>	Contains information about results for a course quiz, including information on each user and quiz question.
<code>report-bulk-meeting</code>	Contains information about all meetings.
<code>report-bulk-meeting-attendance</code>	Contains information about all meeting attendees.
<code>report-bulk-slide-views</code>	Contains slide view data.

Tag	Description
report-course-status	Contains information about the status of a course.
report-disk-usage	Indicates how much hard disk space the Breeze content uses, in bytes.
report-meeting-attendance	Contains one or more <code>row</code> tags, each containing information about a user who attended a particular meeting.
report-meeting-concurrent-users	The maximum number of users who participated simultaneously in the specified meeting.
report-meeting-session	Contains information about a meeting.
report-meeting-sessions	Contains information about all the sessions of a meeting.
report-meeting-session-slots	Contains one or more <code>row</code> tags, each containing information about a particular session slot.
report-meeting-summary	Indicates how many users were invited to the specified meeting and how many invitees and guests actually attended.
report-principal-list	Contains information about the principals.
report-quiz-answer-distribution	Contains one or more <code>row</code> tags, each indicating how many users selected each answer to the specified quiz question.
report-quiz-interactions	Contains one or more <code>row</code> tags, each containing information about an interaction a user had with the specified quiz.
report-quiz-question-answer-distribution	Contains one or more <code>row</code> tags, each indicating how many users selected a particular answer to a particular quiz question.
report-quiz-question-distribution	Contains one or more <code>row</code> tags, each indicating how many users answered a question in the specified quiz correctly.
report-quiz-question-response	Contains one or more <code>row</code> tags, each describing an answer that a user gave to a particular quiz question.
report-quiz-question-totals	Indicates, for the specified quiz question, the total number of users who answered the question and the number of users who answered the question correctly.
report-quiz-results	The number of users who took a specified quiz and their range of scores.
report-quiz-summary	Contains information about a quiz.
report-quiz-takers	Contains one or more <code>row</code> tags, each containing information about a user who has taken the specified course or quiz.
report-quotas	Contains information about the account quotas.
report-sco-slides	Contains one or more <code>row</code> tags, each indicating how many times and how recently a slide in a presentation has been viewed.

Tag	Description
<code>report-sco-views</code>	Contains one or more <code>row</code> tags, each indicating how many times and how recently the specified presentation has been viewed.
<code>report-survey-question-response</code>	Contains information about the list of all users who answered the specified question and their answer.
<code>report-training-concurrent-users</code>	Indicates the largest number of users who have used the system at the same time.
<code>response</code>	A user's response to a survey question.
<code>results</code>	Contains all results returned by any action.
<code>row</code>	Contains a single record of information in a report.
<code>sco</code>	Contains tags that contain information about a SCO, which is a single Breeze document.
<code>sco-author</code>	Contains tags that provide information about the author of a SCO.
<code>sco-data</code>	Contains information about the underlying data file (such as the file that contains a presentation) used in a SCO.
<code>sco-name</code>	The name of a particular SCO, such as the name of a meeting.
<code>sco-nav</code>	Contains tags that contain information about the folder hierarchy that contains the specified SCO.
<code>sco-search-info</code>	Contains one or more <code>sco</code> tags, each containing information about a SCO that matches the specified search text.
<code>scos</code>	Contains one or more <code>sco</code> tags, each containing information about a folder that contains content relevant to the logged-in user.
<code>shortcuts</code>	Contains one or more <code>sco</code> tags, each containing information about a folder that contains content relevant to the logged-in user.
<code>source-sco</code>	Contains information about the SCO for the content inside a course.
<code>status</code>	A status code returned by the Breeze server whenever an action is called.
<code>time-slot-begin</code>	The beginning date and time for a meeting session.
<code>time-slot-end</code>	The ending date and time for a meeting session.
<code>time-taken</code>	The time a course was taken.
<code>tree</code>	Contains a tree that belongs to a specified account.
<code>trees</code>	Contains the list of trees that belong to the specified account.
<code>url</code>	A URL.
<code>url-path</code>	The path on the Breeze server to the specified SCO.

Tag	Description
<code>user (common-info)</code>	Contains information about a user.
<code>user (user-accounts)</code>	Contains information about a user and an account.
<code>user-first-name</code>	A user's first name.
<code>user-last-name</code>	A user's last name.
<code>user-login</code>	A user's login (usually an e-mail address).
<code>user-name</code>	A user name.
<code>users</code>	Contains one or more <code>user</code> tags, each containing information about a user and an account.
<code>value</code>	The value of a custom field.
<code>version</code>	The Breeze version number.
<code>view-date-time</code>	The time and date that a slide was viewed.

access-key (container)

There are two tags named `access-key`; they are described in different sections.

Description

Container tag; contains information about a special access key provided by a SCO.

Attributes

- `acl-id` The ID of the SCO associated with the access key.
- `action` Indicates the type of self-registration. The possible values are "action-self-reg-meeting" and "action-self-reg-course".

Contained tags

`access-key (key)`, `date-created`

Returned by

`accesskey-info`

Example

```
<access-key acl-id="622847" action="action-self-reg-meeting"
  parent-acl-id="503265">
  <access-key>3hzvnifcfmphn3id</access-key>
  <date-created>2004-03-05T10:28:14.750-08:00</date-created>
</access-key>
```

See also

`access-key (key)`, `access-keys`

access-key (key)

There are two tags named `access-key`; they are described in different sections.

Description

A special access key for use in self-registration.

Attributes

None.

Contained tags

None (content only).

Returned by

[accesskey-info](#)

Example

```
<access-key>vb7gmg6ohtz4y7k6</access-key>
```

See also

[access-key \(container\)](#), [access-keys](#)

access-key-group

Description

Container tag; contains information about which group the user automatically joins by self-registering using the special access key. Also contains a login and name tag.

Attributes

`account-id` The ID of the account of which the principal is a member. This attribute is returned only by the `principal-info` action.

`has-children` Indicates whether the principal has children. Groups have children and users don't, so this attribute indicates whether the principal is a group.

`is-primary` Indicates whether the principal is a *primary group*; for more information, see the `type` attribute.

`principal-id` The ID of the principal.

`type` The type of principal. Possible values include "user", "group", and various kinds of *primary groups*: "admins", "authors", "live-admins", and "course-admins". You can create and modify users and groups, but not primary groups.

Contained tags

[login](#), [name](#)

Returned by

[accesskey-info](#)

Example

```
<access-key-group account-id="7" has-children="true" is-primary="false"
  principal-id="630" type="group">
  <login>COURSE-test1-self-reg-group</login>
  <name>COURSE-test1-self-reg-group</name>
</access-key-group>
```

access-keys

Description

Container tag; contains one or more access keys.

Attributes

None.

Contained tags

[access-key](#) (container)

Returned by

[accesskey-info](#)

Example

```
<access-keys>
  <access-key acl-id="619605" action="1" parent-acl-id="619605">
    <access-key>vb7gmg6ohtz4y7k6</access-key>
    <date-created>2004-03-05T00:30:38.483-08:00</date-created>
  </access-key>
</access-keys>
```

account

Description

The account ID.

Attributes

`account-id` The ID of the account.

Contained tags

None (empty tag).

Returned by

[common-info](#)

Example

```
<account account-id="222914"/>
```

action

Description

The name of an action.

Attributes

None.

Contained tags

None (content only).

Returned by

`action-list`

Example

```
<action>user-accounts</action>
```

actions

Description

Container tag; contains a list of actions.

Attributes

None.

Contained tags

`action`

Returned by

`action-list`

Example

```
<actions>
  <action>accesskey-exec</action>
  <action>accesskey-info</action>
  <action>action-list</action>
  ...
  [other actions listed here]
  ...
  <action>user-accounts</action>
  <action>user-update-pwd</action>
</actions>
```

answer-correct

Description

The text for a correct answer.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-content-quiz](#)

Example

```
<answer-correct>YES</answer-correct>
```

answer-text

Description

The text of an answer to a quiz question as part of the results returned by a report action.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-quiz-answer-distribution](#), [report-quiz-question-response](#)

Example

```
<answer-text>ECMA-262</answer-text>
```

body

Description

The text of a returned item, such as the text of a quiz question in a quiz-related report.

Attributes

None.

Contained tags

None (content only).

Returned by

report-quiz-answer-distribution, report-quiz-definition-answers,
report-quiz-definition-questions, report-quiz-question-distribution,
report-quiz-question-response

Example

```
<body>What is 2+2?</body>
```

comment

Description

Any comments on the custom field (may be 0 to 60 characters long).

Attributes

None.

Contained tags

None (content only).

Returned by

custom-field-update

Example

```
<comments>any comment goes here</comments>
```

common

Description

Container tag; contains tags that provide information about the server and the logged-in user.

If you call `common-info` without logging in first, the same information is returned except for the `user` tag.

Attributes

`locale` The locale of the currently logged-in user.

Contained tags

account, cookie, date, host, url, user (`common-info`), version

Returned by

common-info

Example

```
<common locale="en">
  <cookie>breezm5qtgnye46zpckbf.MARIANNE</cookie>
  <date>2004-02-12T15:53:19.797-07:00</date>
  <host>http://admin.breeze.example.com</host>
  <url>/api/xml?accesskey=yyyyyyyyyyyyyyyy&action=common-info</url>
```

```
<version>breeze_402_r116</version>
<account account-id="222914" />
<user user-id="503562">
  <login>gerweck@example.com</login>
</user>
</common>
```

contact

Description

Container tag; contains information about a contact.

Attributes

`contact-id` The ID of a contact.

Contained tags

`email`, `first-name`, `last-name`

Returned by

`principal-info`

Example

```
<contact contact-id="654">
  <email>test4-lnagaraj@test.enang.com</email>
  <first-name>test4</first-name>
  <last-name>laxmi</last-name>
</contact>
```

cookie

Description

The value of the Java servlet session cookie, named BREEZESSESSION.

This is the same cookie that you send with every action call to indicate which user is making the call. If you call `common-info` without logging in first, the server returns a new cookie.

You need the cookie when, for example, you create a new browser window or launch a SCO. Depending on the way your external system is implemented, it may be more convenient to acquire the cookie from the server using `common-info` than for that layer of your code to determine what cookie you're sending when you call an action.

Attributes

None.

Contained tags

None (content only).

Returned by

`common-info`

Example

```
<cookie>breezm5qtgnye46zpckbf.MARIANNE</cookie>
```

course

Description

Container tag; contains information about a course.

Attributes

- `sco-id` The ID of the course.
- `status` Indicates whether a course is active. The possible values are "no-content", which indicates that the course has just been created but has no content, and "active".
- `type` Indicates that this SCO is a course. The value is always "course".

Contained tags

`date-begin`, `date-created`, `date-end`, `date-modified`, `expired`, `name`, `url`, `url-path`

Returned by

`report-my-courses`

Example

```
<course sco-id="555038" status="active" type="course">
  <name>My Name</name>
  <url>admin.breeze.example.com/p38547860</url>
  <date-created>2004-02-09T14:40:38.497-08:00</date-created>
  <date-modified>2004-02-09T14:40:44.623-08:00</date-modified>
  <date-begin>2004-02-09T14:30:00.000-08:00</date-begin>
  <date-end>2004-02-09T14:30:00.000-08:00</date-end>
  <url-path>/p38547860</url-path>
  <expired>true</expired>
</course>
```

custom-fields

Description

Container tag; contains information about the custom fields of an account.

Attributes

None.

Contained tags

`field`

Returned by

`custom-fields`

Example

```
<custom-fields>
  <field field-id="1329" custom-seq="1" account-id="7" type="optional">
    <name>Department</name>
    <comments>this is optional</comments>
  </field>
  <field field-id="1330" custom-seq="2" account-id="7"
type="required">
    <name>SSN</name>
  </field>
</custom-fields>
```

date

Description

The current date and time on the server, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[common-info](#)

Example

```
<date>2004-02-12T15:53:19.797-07:00</date>
```

date-begin

Description

The date and time that a SCO is, or was, scheduled to begin, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-active-meetings](#), [report-meeting-session](#), [report-meeting-sessions](#),
[report-my-courses](#), [report-my-meetings](#), [sco-contents](#)

Example

```
<date-begin>2003-09-02T21:41:58.330-07:00</date-begin>
```

date-closed

Description

The time at which the user closed the window that contains the meeting, thereby leaving the meeting, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-meeting-attendance](#)

Example

```
<date-closed>2004-01-29T00:59:41.470-08:00</date-closed>
```

date-created

Description

The date and time that a SCO was created, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[accesskey-info](#), [report-active-meeting-presenters](#), [report-meeting-attendance](#),
[report-my-courses](#), [report-quiz-interactions](#), [sco-info](#), [sco-search](#)

Example

```
<date-created>2004-02-09T14:40:38.497-08:00</date-created>
```

date-end

Description

The date and time that a SCO is, or was, scheduled to end, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-meeting-session](#), [report-my-courses](#), [report-my-meetings](#), [sco-contents](#)

Example

```
<date-end>2004-02-09T14:30:00.000-08:00</date-end>
```

date-expired

Description

The date that an account is, or was, scheduled to expire, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[user-accounts](#)

Example

```
<date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
```

date-last-taken

Description

The date that a course was last taken, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-course-status](#)

Example

```
<date-last-taken>2004-07-13T10:55:28.763-07:00</date-last-taken>
```

date-modified

Description

The date that something was most recently modified, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-my-courses](#), [report-quiz-definition-questions](#), [sco-contents](#), [sco-info](#), [sco-search](#)

Example

```
<date-modified>2004-02-09T14:40:44.623-08:00</date-modified>
```

date-taken

Description

The date that a particular user took a particular quiz, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-quiz-takers](#)

Example

```
<date-taken>2004-01-13T01:33:49.460-08:00</date-taken>
```

date-time-attempted

Description

The date that a particular user took a particular quiz, in ISO 8601 format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-content-quiz-results](#), [report-bulk-course-results](#)

Example

```
<date-time-attempted>2004-07-13T10:52:23.780-07:00</date-time-attempted>
```

description

Description

The description of a principal.

Attributes

None.

Contained tags

None (content only).

Returned by

[principal-info](#), [principal-list](#)

Example

```
<description>Account administrators group</description>
```

domain-name

Description

The fully qualified domain name of the location of a SCO.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-my-meetings](#), [sco-shortcuts](#)

Example

```
<domain-name>admin.breeze.example.com</domain-name>
```

duration

Description

The duration of a SCO.

Attributes

None.

Contained tags

None (content only).

Returned by

[sco-contents](#)

Example

```
<duration>1900-01-02T/+:00:00.000+00:00</duration>
```

email

Description

A user's e-mail address.

Attributes

None.

Contained tags

None (content only).

Returned by

[principal-list](#), [sco-info](#)

Example

```
<email>tenant@example.com</email>
```

expired

Description

A Boolean value that indicates whether a SCO has expired.

A SCO has *expired* when its end date has passed, and it is no longer available.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-my-courses](#), [report-my-meetings](#)

Example

```
<expired>true</expired>
```

field

Description

Container tag; contains information about a new custom field.

Attributes

`field-id` The ID of the field.

`account-id` The ID of the account.

custom-seq The custom field number.

type The type of the custom field. It must be one of the following values:

Value	Description
required	Required account custom field
optional	Optional and show during self-registration
optional-no-self-reg	Optional and hide during self-registration

Contained tags

[comment](#), [name](#), [value](#)

Note: Not all tags are returned by all actions.

Returned by

[custom-field-update](#)

Example

```
<field field-id="1562" account-id="7" custom-seq="4"
  type="optional">
  <comments>none</comments>
  <name>Manager</name>
  <value>411</value>
</field>
```

first-name

Description

A user's first name.

Attributes

None.

Contained tags

None (content only).

Returned by

[sco-info](#), [sco-search](#)

Example

```
<first-name>Pat</first-name>
```

hit

Description

A number indicating which slide contains a search term.

Attributes

None.

Contained tags

None (content only).

Returned by

`sco-search`

Example

```
<hit>5</hit>
```

host

Description

The URL of the host that the Breeze server is running on.

Attributes

None.

Contained tags

None (content only).

Returned by

`common-info`

Example

```
<host>http://admin.breeze.example.com</host>
```

invalid

Description

Container tag; contains information about why a particular call to an action is invalid.

This element appears only inside a `status` tag and only when the `code` attribute of that tag is set to "invalid".

The server returns only one `invalid` tag for a given request, even if you leave out multiple required parameters. To resolve a missing-parameter error, add the required parameter to the action. For information on which parameters are required, see the reference material for the specific action in [Chapter 4, "Action Reference," on page 37](#).

Attributes

field The name of the parameter that was invalid.

subcode The reason that the parameter was invalid. The following table shows the possible values for this attribute:

Value	Description
duplicate	The call attempted to add a duplicate item in a context where uniqueness is required.
format	A passed parameter had the wrong format.
illegal-operation	The requested operation violates integrity rules (for example, moving a folder into itself is an illegal operation).
missing	A required parameter was missing.
no-such-item	The requested information does not exist.
range	The value is outside the permitted range of values.

type The type of the parameter, such as "id", "long", or "string".

Contained tags

None (empty tag).

Returned by

Any invalid action.

Example

```
<invalid field="sco-id" type="id" subcode="missing" />
```

last-name

Description

A user's last name.

Attributes

None.

Contained tags

None (content only).

Returned by

[sco-info](#), [sco-search](#)

Example

```
<last-name>Tennant</last-name>
```

last-viewed

Description

The date and time of the most recent time that a SCO was viewed, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-sco-slides](#)

Example

```
<last-viewed>2004-01-13T16:35:39.517-08:00</last-viewed>
```

last-viewed-date

Description

The most recent date and time that the SCO was viewed, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-sco-views](#)

Example

```
<last-viewed-date>2004-01-13T17:11:47.923-08:00</last-viewed-date>
```

login

Description

A user's login name, which is usually the user's e-mail address.

Attributes

None.

Contained tags

None (content only).

Returned by

[common-info](#), [permissions-info](#), [principal-info](#), [principal-list](#), [report-course-takers](#), [report-meeting-attendance](#), [report-quiz-takers](#)

Example

```
<login>tenant@example.com</login>
```

meeting

Description

Container tag; contains information about a meeting.

Attributes

`active-participants` The number of current participants.

`permission-id` The permissions the user has for the meeting. For more information, see [“About security” on page 17](#).

`sco-id` The ID of the meeting.

`status` Indicates the meeting’s status. The possible values are "no-content", which indicates that the meeting has just been created but has no content; "in-progress", which indicates a meeting that’s currently in progress; and "active".

`type` Indicates that this SCO is a meeting. The value is always "meeting".

Contained tags

[date-begin](#), [date-end](#), [domain-name](#), [expired](#), [name](#), [url-path](#)

Returned by

[report-my-meetings](#)

Example

```
<meeting sco-id="181487" type="meeting" status="active" permission-id="view"
  length="36000000" active-participants="0">
  <name>morris shared</name>
  <domain-name>admin.breeze.example.com</domain-name>
  <url-path>/r99718924/</url-path>
  <date-begin>2003-07-14T23:00:00.000-07:00</date-begin>
  <date-end>2003-07-15T09:00:00.000-07:00</date-end>
  <expired>true</expired>
</meeting>
```

my-courses

Description

Container tag; contains information about one or more courses.

Attributes

None.

Contained tags

[course](#)

Returned by

[report-my-courses](#)

Example

```
<my-courses>
  <course sco-id="555038" status="active" type="course">
    <name>My Name</name>
    <url>admin.breeze.example.com/p38547860</url>
    <date-created>2004-02-09T14:40:38.497-08:00</date-created>
    <date-modified>2004-02-09T14:40:44.623-08:00</date-modified>
    <date-begin>2004-02-09T14:30:00.000-08:00</date-begin>
    <date-end>2004-02-09T14:30:00.000-08:00</date-end>
    <url-path>p38547860</url-path>
    <expired>true</expired>
  </course>
</my-courses>
```

my-meetings

Description

Container tag; contains information about one or more meetings.

Attributes

None.

Contained tags

[meeting](#)

Returned by

[report-my-meetings](#)

Example

```
<my-meetings>
  <meeting sco-id="181487" type="meeting" status="active" permission-id="view"
    length="36000000" active-participants="0">
    <name>morris shared</name>
    <domain-name>admin.breeze.example.com</domain-name>
    <url-path>/r99718924</url-path>
    <date-begin>2003-07-14T23:00:00.000-07:00</date-begin>
    <date-end>2003-07-15T09:00:00.000-07:00</date-end>
    <expired>true</expired>
  </meeting>
</my-meetings>
```

name

Description

A user's name. For example, depending on the context, the tag may contain the name of a person, the name of a group, the name of a SCO, an e-mail address, or another kind of name.

Attributes

None.

Contained tags

None (content only).

Returned by

[permissions-info](#), [principal-info](#), [principal-list](#), [report-active-meeting-presenters](#), [report-active-meetings](#), [report-meeting-session](#), [report-meeting-sessions](#), [report-my-courses](#), [report-my-meetings](#), [report-quiz-takers](#), [report-sco-views](#), [report-training-concurrent-users](#), [sco-contents](#), [sco-info](#), [sco-nav](#), [sco-search](#), [user-accounts](#)

Example

```
<name>Math Test</name>
```

participant-name

Description

The name of a participant in a meeting.

The name is presented in the standard format for the locale.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-meeting-attendance](#)

Example

```
<participant-name>Tennant, Pat</participant-name>
```

permission

Description

Contains information about a specific principal and its permissions.

Attributes

`acl-id` The ID of a specific SCO.

`permission-id` The permissions that the principal has for the specified SCO. This attribute is returned only by the `permissions-info` action.

`principal-id` The ID of the principal.

Contained tags

None (empty tag).

Returned by

`permissions-info`

Example

```
<permission acl-id="27" permission-id="manage" principal-id="8" />
```

permissions

Description

Container tag; contains information about one or more principals and their permissions.

For information about principals, see [“About principals, SCOs, and IDs” on page 16](#). For information about permissions, see [“About security” on page 17](#).

Attributes

None.

Contained tags

`principal`

Returned by

`permissions-info`

Example

```
<permissions>
  <principal principal-id="181248" is-primary="false" type="user"
    has-children="false" permission-id="view">
    <name>Englesberg, Ari</name>
    <login>englesberg@example.com</login>
  </principal>
</permissions>
```

physical-path

Description

The path on the server to a SCO.

Attributes

None.

Contained tags

None (content only).

Returned by

sco-info

Example

<physical-path>222914/503227/</physical-path>

preferences

Description

Contains information about user preferences.

Attributes

acl-id The ID of the user for whom the preferences are updated.

lang The language setting. Breeze currently supports the following five languages:

Language	Parameter value
English	en
French	fr
German	de
Japanese	ja
Korean	ko

time-zone-id The time zone setting. Breeze currently supports the following 75 time zones:

Time zone setting	Parameter value
International Date Line West (GMT-12:00)	0
Midway Island, Samoa (GMT-11:00)	1
Hawaii (GMT-10:00)	2
Alaska (GMT-09:00)	3
Pacific Time (US and Canada); Tijuana (GMT-08:00)	4
Mountain Time (US and Canada) (GMT-07:00)	10

Time zone setting	Parameter value
Chihuahua, La Paz, Mazatlan (GMT-07:00)	13
Arizona (GMT-07:00)	15
Central Time (US and Canada) (GMT-06:00)	20
Saskatchewan (GMT-06:00)	25
Guadalajara, Mexico City, Monterrey (GMT-06:00)	30
Central America (GMT-06:00)	33
Eastern Time (US and Canada) (GMT-05:00)	35
Indiana (East) (GMT-05:00)	40
Bogota, Lima, Quito (GMT-05:00)	45
Atlantic Time (Canada) (GMT-04:00)	50
Caracas, La Paz (GMT-04:00)	55
Santiago (GMT-04:00)	56
Newfoundland (GMT-03:30)	60
Brasilia (GMT-03:00)	65
Buenos Aires, Georgetown (GMT-03:00)	70
Greenland (GMT-03:00)	73
Mid-Atlantic (GMT-02:00)	75
Azores (GMT-01:00)	80
Cape Verde Islands (GMT-01:00)	83
Dublin, Edinburgh, Lisbon, London (GMT)	85
Casablanca, Monrovia (GMT)	90
Belgrade, Bratislava, Budapest, Ljubljana, Prague (GMT+01:00)	95
Sarajevo, Skopje, Warsaw, Zagreb (GMT+01:00)	100
Brussels, Copenhagen, Madrid, Paris (GMT+01:00)	105
Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna (GMT+01:00)	110
West Central Africa (GMT+01:00)	113
Bucharest (GMT+02:00)	115
Cairo (GMT+02:00)	120
Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius (GMT+02:00)	125
Athens, Istanbul, Minsk (GMT+02:00)	130
Jerusalem (GMT+02:00)	135
Harare, Pretoria (GMT+02:00)	140
Moscow, St. Petersburg, Volgograd (GMT+03:00)	145

Time zone setting	Parameter value
Kuwait, Riyadh (GMT+03:00)	150
Nairobi (GMT+03:00)	155
Baghdad (GMT+03:00)	158
Tehran (GMT+03:30)	160
Abu Dhabi, Muscat (GMT+04:00)	165
Baku, Tbilisi, Yerevan (GMT+04:00)	170
Kabul (GMT+04:30)	175
Ekaterinburg (GMT+05:00)	180
Islamabad, Karachi, Tashkent (GMT+05:00)	185
Chennai, Kolkata, Mumbai, New Delhi (GMT+05:30)	190
Kathmandu (GMT+05:45)	193
Astana, Dhaka (GMT+06:00)	195
Sri Jayawardenepura (GMT+06:00)	200
Almaty, Novosibirsk (GMT+06:00)	201
Rangoon (GMT+06:30)	203
Bangkok, Hanoi, Jakarta (GMT+07:00)	205
Krasnoyarsk (GMT+07:00)	207
Beijing, Chongqing, Hong Kong SAR, Urumqi (GMT+08:00)	210
Kuala Lumpur, Singapore (GMT+08:00)	215
Taipei (GMT+08:00)	220
Perth (GMT+08:00)	225
Irkutsk, Ulaan Bataar (GMT+08:00)	227
Seoul (GMT+09:00)	230
Osaka, Sapporo, Tokyo (GMT+09:00)	235
Yakutsk (GMT+09:00)	240
Darwin (GMT+09:30)	245
Adelaide (GMT+09:30)	250
Canberra, Melbourne, Sydney (GMT+10:00)	255
Brisbane (GMT+10:00)	260
Hobart (GMT+10:00)	265
Vladivostok (GMT+10:00)	270
Guam, Port Moresby (GMT+10:00)	275
Magadan, Solomon Islands, New Caledonia (GMT+11:00)	280

Time zone setting	Parameter value
Fiji Islands, Kamchatka, Marshall Islands (GMT+12:00)	285
Auckland, Wellington (GMT+12:00)	290
Nuku'alofa (GMT+13:00)	300

Contained tags

None.

Returned by

[principal-info](#)

Example

```
<preferences acl-id="653" lang="en" time-zone-id="4" />
```

presentation-name

Description

The name of a Breeze presentation.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-content-quiz](#), [report-bulk-slide-views](#)

Example

```
<presentation-name>test-quiz</presentation-name>
```

principal

Description

Contains information about a principal.

For general information about principals, see [“About principals, SCOs, and IDs” on page 16](#).

Attributes

account-id The ID of the account of which the principal is a member. This attribute is returned only by the [principal-info](#) action.

has-children Indicates whether the principal has children. Groups have children and users don't, so this attribute indicates whether the principal is a group.

is-primary Indicates whether the principal is a primary group; for more information, see the [type attribute](#).

`permission-id` The permissions that the principal has for the specified SCO. This attribute is returned only by the `permissions-info` action.

`principal-id` The ID of the principal.

`contact-id` The ID for the contact information associated with the user. This value is `null` for groups.

`type` The type of principal. Possible values include "user", "group", and various kinds of primary groups: "admins", "authors", "live-admins", and "course-admins". You can create and modify users and groups, but not primary groups.

Contained tags

`field`, `name`,

Returned by

`permissions-info`, `principal-update`

Example

```
<principal-custom-field-values>
  <field field-id="652" custom-seq="1" account-id="7" type="optional"
principal-id="653">
    <name>Phone number</name>
    <value>408-1111-111</value>
  </field>
</principal-custom-field-values>
```

principal-custom-field-values

Description

Container tag; contains information about the values of a custom field in a principal.

For general information about principals, see [“About principals, SCOs, and IDs” on page 16](#).

Attributes

None.

Contained tags

`field`

Returned by

`principal-info`

Example

```
<principal-custom-field-values>
  <field field-id="652" custom-seq="1" account-id="7" type="optional"
principal-id="653">
    <name>Phone number</name>
    <value>408-1111-111</value>
  </field>
</principal-custom-field-values>
```

principal-list

Description

Container tag; contains one or more principal tags.

Attributes

None.

Contained tags

[principal](#)

Returned by

[principal-list](#)

Example

```
<principal-list>
  <principal principal-id="222917" type="admins" has-children="true"
    is-primary="true" is-hidden="0">
    <name>Account Administrators</name>
    <login>Account Administrators</login>
    <description>Account administrators group</description>
  </principal>
</principal-list>
```

principal-name

Description

The name of a principal.

The name is presented in the standard format for the locale.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-course-takers](#), [report-quiz-takers](#)

Example

```
<principal-name>Chiang, Ed</principal-name>
```

question

Description

The text of a quiz question.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-quiz-interactions](#), [report-survey-question-response](#)

Example

```
<question>What is 2+2?</question>
```

question-text

Description

The text of a quiz question.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-quiz-question-answer-distribution](#), [report-bulk-content-quiz](#)

Example

```
<question-text>What is 2+2?</question-text>
```

quiz-name

Description

The name of a quiz.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-content-quiz](#)

Example

```
<question-text>Who is Randy Johnson ?</question-text>
```

quiz-definition-answers

Description

Container tag; contains one or more `row` tags, each containing information about a particular answer to a particular quiz question.

Attributes

None.

Contained tags

`row`

Returned by

`report-quiz-definition-answers`

Example

```
<quiz-definition-answers>
  <row answer-id="1" question-id="257" correct="0">
    <body>1</body>
  </row>
  <row answer-id="2" question-id="257" correct="1">
    <body>2</body>
  </row>
  <row answer-id="1" question-id="258" correct="1">
    <body>1</body>
  </row>
  <row answer-id="2" question-id="258" correct="0">
    <body>2</body>
  </row>
</quiz-definition-answers>
```

quiz-definition-questions

Description

Container tag; contains one or more `row` tags, each containing information about a particular quiz question.

Attributes

None.

Contained tags

`row`

Returned by

`report-quiz-definition-questions`

Example

```
<quiz-definition-questions>
  <row question-number="1" question-id="257" score="20" chapter-id="1">
    <body>What is 2+2?</body>
    <date-modified>2004-01-13T17:09:40.230-08:00</date-modified>
  </row>
  <row question-number="2" question-id="258" score="20" chapter-id="1">
    <body>How far is it from Earth to the sun?</body>
    <date-modified>2004-01-13T17:09:40.230-08:00</date-modified>
  </row>
</quiz-definition-questions>
```

quiz-name

Description

The name of a quiz.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-content-quiz](#)

Example

```
<quiz-name>Quiz test 1</quiz-name>
```

quota

Description

Container tag; contains information on the account quotas.

Attributes

`acl-id` The ID of the SCO associated with the access key.

`quota-id` The ID of a quota. Breeze enforces different quotas, for example, the Training concurrent user peak, the download used, the authors used, and so on. Each quota has an associated unique quota ID.

`used` The amount of quota that was used. For example, if the account allows a maximum of five authors, and three authors exist in the account, the value of `used` would be 3.

`limit` The original limit for the quota. This value is used for load balancing.

`soft-limit` The elevated limit for the quota. This value is used for the Breeze pricing models that allow customers to quickly scale their license beyond their limit.

Contained tags

[date-begin](#), [date-end](#)

Returned by

[report-quotas](#)

Example

```
<quota acl-id="7" quota-id="2" used="0" limit="unlimited" soft-
  limit="1000000000">
  <date-begin>1999-12-31T16:00:00.000-08:00</date-begin>
  <date-end>2999-12-31T16:00:00.000-08:00</date-end>
</quota>
```

report-account-meeting-attendance

Description

Container tag; contains rows tags that contain information about the meeting attendance log for the account.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-account-meeting-attendance](#)

Example

```
<report-account-meeting-attendance>
  <row account-id="7" transcript-id="670" sco-id="623" principal-id="8"
    contact-id="3">
    <login>lnagaraj@macromedia.com</login>
    <session-name>Enterprise Administrator</session-
      name>
    <sco-name>test-mtg-1</sco-name>
    <date-created>2004-07-09T13:00:28.920-07:00</date-
      created>
    <date-closed>2004-07-09T13:02:55.810-07:00</date-
      closed>
    <participant-name>Enterprise Administrator</participant-
      name>
  </row>
  <row account-id="7" transcript-id="685" sco-id="623" principal-id="8"
    contact-id="3">
    <login>lnagaraj@macromedia.com</login>
    <session-name>Enterprise Administrator</session-
      name>
    <sco-name>test-mtg-1</sco-name>
    <date-created>2004-07-09T14:33:06.577-07:00</date-
      created>
```

```

        <date-closed>2004-07-09T14:33:43.547-07:00</date-
closed>
        <participant-name>Enterprise Administrator</participant-
name>
    </row>
</report-account-meeting-attendance>

```

report-active-meeting-presenters

Description

Container tag; contains a list of the users who are currently presenting meetings.

Attributes

None.

Contained tags

[sco](#)

Returned by

[report-active-meeting-presenters](#)

Example

```

<report-active-meeting-presenters>
  <sco sco-id="190298">
    <name>Jack Monson</name>
    <date-created>2004-02-17T17:06:23.920-08:00</date-created>
  </sco>
  <sco sco-id="566058">
    <name>Hans Gay</name>
    <date-created>2004-02-17T16:50:00.327-08:00</date-created>
  </sco>
</report-active-meeting-presenters>

```

report-active-meetings

Description

Container tag; contains a list of meetings that are currently in progress.

Attributes

None.

Contained tags

[sco](#)

Returned by

[report-active-meetings](#)

Example

```

<report-active-meetings>

```

```

<sco sco-id="196311" active-participants="1" length-minutes="241656">
  <name>QT load</name>
  <date-begin>2003-09-02T21:41:58.330-07:00</date-begin>
</sco>
<sco sco-id="184847" active-participants="5" length-minutes="292290">
  <name>Take 2</name>
  <date-begin>2003-07-29T17:48:25.273-07:00</date-begin>
</sco>
</report-active-meetings>

```

report-bandwidth

Description

The total bandwidth consumed in the current account since the account was created, in bytes.

Attributes

total The total amount of bandwidth, in bytes.

Contained tags

None (empty tag).

Returned by

[report-bandwidth](#)

Example

```
<report-bandwidth total="71000000" />
```

report-bulk-content-quiz

Description

Container tag; contains rows that contain information about all content quizzes, including a list of quizzes.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-content-quiz](#)

Example

```

<report-bulk-content-quiz>
  <row presentation-id="720" quiz-id="1" quiz-passing-score="10" question-
    number="1" question-value="10" answer-id="1">
    <presentation-name>test-quiz</presentation-name>
    <quiz-name>Quiz test 1</quiz-name>
    <question-text>Who is Randy Johnson ?</question-text>
  </row>
</report-bulk-content-quiz>

```

```

    <answer-text>A baseball pitcher with the Arizona Diamondbacks</answer-
text>
    <answer-correct>YES</answer-correct>
</row>
<row presentation-id="720" quiz-id="1" quiz-passing-score="10" question-
number="1" question-value="10" answer-id="2">
    <presentation-name>test-quiz</presentation-name>
    <quiz-name>Quiz test 1</quiz-name>
    <question-text>Who is Randy Johnson ?</question-text>
    <answer-text>A basketball player with the New Jersey Nets</answer-text>
    <answer-correct>NO</answer-correct>
</row>
</report-bulk-content-quiz>

```

report-bulk-content-quiz-results

Description

Container tag; contains row information about the results for a content quiz.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-content-quiz-results](#)

Example

```

<report-bulk-content-quiz-results>
  <row user-id="8" presentation-attempt-id="725" presentation-id="720" quiz-
id="1" question-number="1" user-response="1">
    <user-login>lnagaraj@macromedia.com</user-login>
    <date-time-attempted>2004-07-13T10:51:35.047-07:00</date-time-attempted>
    <time-taken>00:00:28.017</time-taken>
  </row>
  <row user-id="8" presentation-attempt-id="726" presentation-id="720" quiz-
id="1" question-number="2" user-response="3">
    <user-login>lnagaraj@macromedia.com</user-login>
    <date-time-attempted>2004-07-13T10:52:23.780-07:00</date-time-attempted>
    <time-taken>00:00:14.017</time-taken>
  </row>
</report-bulk-content-quiz-results>

```

report-bulk-course-quiz

Description

Container tag; contains rows that contain information about all course quizzes, including a list of quizzes.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-course-quiz](#)

Example

```
<report-bulk-course-quiz>
  <row course-id="727" quiz-id="1" quiz-passing-score="10" question-
number="1" question-value="10" answer-id="1">
    <presentation-name>quiz-course-1</presentation-name>
    <quiz-name>Quiz test 1</quiz-name>
    <question-text>Who is Randy Johnson ?</question-text>
    <answer-text>A baseball pitcher with the Arizona Diamondbacks</answer-
text>
    <answer-correct>YES</answer-correct>
  </row>
  <row course-id="727" quiz-id="1" quiz-passing-score="10" question-
number="1" question-value="10" answer-id="2">
    <presentation-name>quiz-course-1</presentation-name>
    <quiz-name>Quiz test 1</quiz-name>
    <question-text>Who is Randy Johnson ?</question-text>
    <answer-text>A basketball player with the New Jersey Nets</answer-text>
    <answer-correct>NO</answer-correct>
  </row>
</report-bulk-course-quiz>
</results>
```

report-bulk-course-quiz-results

Description

Container tag; contains information about results for a course quiz, including information on each user and quiz question.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-course-results](#)

Example

```
<report-bulk-course-quiz-results>
  <row user-id="8" course-attempt-id="730" course-id="727" quiz-id="1"
    question-number="1" user-response="2">
    <user-login>lnagaraj@macromedia.com</user-login>
    <date-time-attempted>2004-07-13T10:54:09.030-07:00</date-time-attempted>
    <time-taken>00:00:43.017</time-taken>
  </row>
  <row user-id="8" course-attempt-id="730" course-id="727" quiz-id="1"
    question-number="2" user-response="1">
    <user-login>lnagaraj@macromedia.com</user-login>
    <date-time-attempted>2004-07-13T10:54:09.030-07:00</date-time-attempted>
    <time-taken>00:00:67.017</time-taken>
  </row>
</report-bulk-course-quiz-results>
```

report-bulk-meeting

Description

Container tag; contains information about all meetings.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-meeting](#)

Example

```
<report-bulk-meeting>
  <row meeting-id="623" invited-participants="3">
    <meeting-name>test-mtg-1</meeting-name>
    <meeting-scheduled-date-time>2004-07-07T10:00:00.000-07:00</meeting-
scheduled-date-time>
  </row>
  <row meeting-id="679" invited-participants="2">
    <meeting-name>test-mtg-2</meeting-name>
    <meeting-scheduled-date-time>2004-07-09T14:00:00.000-07:00</meeting-
scheduled-date-time>
  </row>
</report-bulk-meeting>
```

report-bulk-meeting-attendance

Description

Container tag; contains information about all meeting attendees.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-meeting-attendance](#)

Example

```
<report-bulk-meeting-attendance>
  <row meeting-id="623" meeting-attendee="8">
    <meeting-name>test-mtg-1</meeting-name>
    <attendee>Enterprise Administrator</attendee>
    <attendee-join>2004-07-09T13:00:28.920-07:00</attendee-join>
    <attendee-exit>2004-07-09T13:02:55.810-07:00</attendee-exit>
  </row>
  <row meeting-id="623" meeting-attendee="8">
    <meeting-name>test-mtg-1</meeting-name>
    <attendee>Enterprise Administrator</attendee>
    <attendee-join>2004-07-09T14:33:06.577-07:00</attendee-join>
    <attendee-exit>2004-07-09T14:33:43.547-07:00</attendee-exit>
  </row>
</report-bulk-meeting-attendance>
```

report-bulk-slide-views

Description

Container tag; contains slide view data.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-bulk-content-slide-views](#)

Example

```
<report-bulk-slide-views>
  <row session-id="10000022" viewing-session="32" presentation-id="27" slide-
    number="1">
    <user-login>lnagaraj@macromedia.com</user-login>
    <user-first-name>Enterprise</user-first-name>
    <user-last-name>Administrator</user-last-name>
    <presentation-name>test-slide</presentation-name>
    <view-date-time>2004-06-30T11:46:10.280-07:00</view-date-time>
  </row>
  <row session-id="10000022" viewing-session="32" presentation-id="27" slide-
    number="2">
    <user-login>lnagaraj@macromedia.com</user-login>
    <user-first-name>Enterprise</user-first-name>
    <user-last-name>Administrator</user-last-name>
    <presentation-name>test-slide</presentation-name>
    <view-date-time>2004-06-30T11:46:12.500-07:00</view-date-time>
  </row>
</report-bulk-slide-views>
```

report-course-status

Description

Container tag; contains information about the status of a course.

Attributes

total-course-completions The total number of courses completed.

total-unique-course-completions The total number of unique courses completed.

num-passed The number of courses passed.

num-failed The number of courses failed.

num-enrollees The number of enrollees in a course.

Contained tag

[date-last-taken](#)

Returned by

[report-course-status](#)

Example

```
<report-course-status total-course-completions="2" total-unique-course-
  completions="1" num-passed="1" num-failed="1" num-enrollees="3">
  <date-last-taken>2004-07-13T10:55:28.763-07:00</date-last-taken>
</report-course-status>
```

report-disk-usage

Description

Indicates how much hard disk space the Breeze content uses, in bytes. Includes all content and archives.

Attributes

`total` The total amount of disk usage, in bytes.

Contained tags

None (empty tag).

Returned by

`report-disk-usage`

Example

```
<report-disk-usage total="849327881" />
```

report-meeting-attendance

Description

Container tag; contains one or more `row` tags, each containing information about a user who attended a particular meeting.

Attributes

None.

Contained tags

`row`

Returned by

`report-meeting-attendance`

Example

```
<report-meeting-attendance>
  <row transcript-id="462409" sco-id="505267" principal-id="181255"
    answered-survey="0">
    <login>tenant@example.com</login>
    <sco-name>My Meeting</sco-name>
    <date-created>2004-01-29T00:59:15.490-08:00</date-created>
    <date-closed>2004-01-29T00:59:41.470-08:00</date-closed>
    <participant-name>Tennant, Pat</participant-name>
  </row>
</report-meeting-attendance>
```

report-meeting-concurrent-users

Description

The maximum number of users who participated simultaneously in the specified meeting.

This is different from the number of users who can participate simultaneously in a meeting because the quota may not have been reached. For example, an account may allow 125 simultaneous users for a meeting but only have 2.

Attributes

`max-participants-freq` The number of times in the current month that the maximum number of users has been reached.

`max-users` The maximum number of users.

Contained tags

None (empty tag).

Returned by

[report-meeting-concurrent-users](#)

Example

```
<results>
<status code="ok" />
<report-meeting-concurrent-users max-users="2" max-participants-freq="5" />
```

report-meeting-session

Description

Container tag, contains information about a meeting.

Attributes

`num-guests` The number of participants who entered the meeting as guests rather than registered attendees.

`num-participants` The total number of participants in the meeting.

`sco-id` The ID of the parent meeting.

`session-sco-id` A unique identifier for the individual session. This value matches the value of the `sco-id` that is passed as a parameter to the [report-meeting-session](#) action.

Contained tags

[date-begin](#), [date-end](#), [name](#)

Returned by

[report-meeting-session](#)

Example

```
<report-meeting-session session-sco-id="505267" sco-id="181576"
  num-participants="0" num-guests="0">
  <name>MyMeeting</name>
  <date-begin>2004-01-29T00:30:00.000-08:00</date-begin>
  <date-end>2004-01-29T00:30:00.000-08:00</date-end>
</report-meeting-session>
```

report-meeting-sessions

Description

Container tag; contains information about all the sessions of a meeting.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-meeting-sessions](#)

Example

```
<report-meeting-sessions>
  <row sco-id="622852" num-participants="1" num-guests="0" num-polls="0">
    <name>My Meeting</name>
    <date-begin>2004-03-05T10:33:09.030-08:00</date-begin>
  </row>
</report-meeting-sessions>
```

report-meeting-session-slots

Description

Container tag; contains one or more [row](#) tags, each containing information about a particular session slot.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-meeting-session-slots](#)

Example

```
<report-meeting-session-slots>
  <row num-attendees="0">
    <time-slot-begin>2004-02-14T06:30:00.000-08:00</time-slot-begin>
    <time-slot-end>2004-02-14T06:40:00.000-08:00</time-slot-end>
  </row>
  <row num-attendees="0">
    <time-slot-begin>2004-02-14T06:40:00.000-08:00</time-slot-begin>
    <time-slot-end>2004-02-14T06:50:00.000-08:00</time-slot-end>
  </row>
</report-meeting-session-slots>
```

report-meeting-summary

Description

Indicates how many users were invited to the specified meeting and how many invitees and guests actually attended.

Attributes

num-guests-attended The number of participants who entered the meeting as guests rather than registered attendees.

num-invitees The number of users who were invited.

num-invitees-attended The number of invited users who attended.

Contained tags

None (empty tag).

Returned by

[report-meeting-summary](#)

Example

```
<report-meeting-summary num-invitees="10" num-invitees-attended="5"
  num-guests-attended="1" />
```

report-principal-list

Description

Container tag; contains information about the principals.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-principal-list](#)

Example

```
<report-principal-list>
  <row principal-id="8" type="user" has-children="false" is-primary="false">
    <name>Enterprise Administrator</name>
    <login>lnagaraj@macromedia.com</login>
    <custom0 />
    <custom1 />
  </row>
  <row principal-id="17" type="admins" has-children="true" is-primary="true">
    <name>Account Administrators</name>
    <login>Account Administrators</login>
    <custom0 />
    <custom1 />
  </row>
  <row principal-id="123" type="user" has-children="false" is-primary="false">
    <name>test1 laxmi</name>
    <login>test1-laxmi@test.enang.com</login>
    <custom0>ENGG</custom0>
    <custom1>100-07-7777</custom1>
  </row>
</report-principal-list>
```

report-quiz-answer-distribution

Description

Container tag; contains one or more `row` tags, each indicating how many users selected each answer to the specified quiz question.

Attributes

None.

Contained tags

`row`

Returned by

`report-quiz-answer-distribution`

Example

```
<report-quiz-answer-distribution>
  <row question-number="1" question-id="257" a1="0" a2="0" a3="5" a4="3"
    total="8">
    <body>What is 2+2?</body>
  </row>
  <row question-number="2" question-id="258" a1="8" a2="0" a3="0" a4="0"
    total="8">
    <body>How far is it from Earth to the sun?</body>
  </row>
  <row question-number="3" question-id="259" a1="8" a2="0" a3="0" a4="0"
    total="8">
    <body>Which international standard defines ECMAScript?</body>
  </row>
</report-quiz-answer-distribution>
```

report-quiz-interactions

Description

Container tag; contains one or more `row` tags, each containing information about an interaction a user had with the specified quiz. For more information about interactions, see the [report-quiz-interactions](#) action.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-quiz-interactions](#)

Example

```
<report-quiz-interactions>
  <row transcript-id="459350" question-id="257" sco-id="217706" score="20">
    <date-created>2004-01-13T02:00:46.163-08:00</date-created>
    <question>What is 2+2?</question>
  </row>
  <row transcript-id="459350" question-id="258" sco-id="217706" score="0">
    <date-created>2004-01-13T02:00:46.163-08:00</date-created>
    <question>How far is it from Earth to the sun?</question>
  </row>
  <row transcript-id="459350" question-id="259" sco-id="217706" score="20">
    <date-created>2004-01-13T02:00:46.163-08:00</date-created>
    <question>Which international standard defines ECMAScript?</question>
  </row>
</report-quiz-interactions>
```

report-quiz-question-answer-distribution

Description

Container tag; contains one or more `row` tags, each indicating how many users selected a particular answer to a particular quiz question.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-quiz-question-answer-distribution](#)

Example

```
<report-quiz-question-answer-distribution>
  <row question-id="259" answer-id="1" correct="0" total-responses="1"
    percentage-responded="10">
    <question-text>Which international standard defines
      ECMAScript?</question-text>
    <answer-text>ISO-9000</answer-text>
  </row>
  <row question-id="259" answer-id="2" correct="0" total-responses="3"
    percentage-responded="30">
    <question-text>Which international standard defines
      ECMAScript?</question-text>
    <answer-text>RFC 821</answer-text>
  </row>
  <row question-id="259" answer-id="3" correct="0" total-responses="2"
    percentage-responded="20">
    <question-text>Which international standard defines
      ECMAScript?</question-text>
    <answer-text>ECMA-976</answer-text>
  </row>
  <row question-id="259" answer-id="4" correct="1" total-responses="4"
    percentage-responded="40">
    <question-text>Which international standard defines
      ECMAScript?</question-text>
    <answer-text>ECMA-262</answer-text>
  </row>
</report-quiz-question-answer-distribution>
```

report-quiz-question-distribution

Description

Container tag; contains one or more `row` tags, each indicating how many users answered a question in the specified quiz correctly.

Attributes

None.

Contained tags

`row`

Returned by

`report-quiz-question-distribution`

Example

```
<report-quiz-question-distribution>
  <row question-id="257" question-number="1" num-correct="8" num-incorrect="0"
    percentage-correct="100" score="20">
    <body>What is 2+2?</body>
  </row>
  <row question-id="258" question-number="2" num-correct="8" num-incorrect="0"
    percentage-correct="100" score="20">
    <body>How far is it from Earth to the sun?</body>
```

```

</row>
<row question-id="259" question-number="3" num-correct="4" num-incorrect="6"
  percentage-correct="40" score="20">
  <body>Which international standard defines ECMAScript?</body>
</row>
</report-quiz-question-distribution>

```

report-quiz-question-response

Description

Container tag; contains one or more `row` tags, each describing an answer that a user gave to a particular quiz question.

Attributes

None.

Contained tags

`row`

Returned by

`report-quiz-question-response`

Example

```

<report-quiz-question-response>
  <row principal-id="506299" question-id="257" answer-id="2">
    <body>What is 2+2?</body>
    <answer-text>5</answer-text>
  </row>
  <row principal-id="557527" question-id="257" answer-id="2">
    <body>What is 2+2?</body>
    <answer-text>5</answer-text>
  </row>
  <row principal-id="181255" question-id="257" answer-id="3">
    <body>What is 2+2?</body>
    <answer-text>4</answer-text>
  </row>
</report-quiz-question-response>

```

report-quiz-question-totals

Description

Indicates, for the specified quiz question, the total number of users who answered the question and the number of users who answered the question correctly.

Attributes

`num-correct-answers` The number of users who answered the question correctly.

`question-id` The ID of the question.

`total-responses` The total number of users who answered the question.

Contained tags

None (empty tag).

Returned by

[report-quiz-question-totals](#)

Example

```
<report-quiz-question-totals question-id="260" total-responses="8"
  num-correct-answers="5" />
```

report-quiz-results

Description

The number of users who took a specified quiz and their range of scores.

Attributes

`average-score` The average of the scores of the users who took the quiz.

`high-score` The highest score.

`low-score` The lowest score

`numdistincttaken` The number of distinct users who have taken the quiz.

`numtaken` The total number of times the quiz has been taken.

Contained tags

None (empty tag).

Returned by

[report-quiz-summary](#)

Example

```
<report-quiz-results average-score="95" low-score="90" high-score="100"
  numtaken="8" numdistincttaken="1" />
```

report-quiz-summary

Description

Contains information about a quiz.

Attributes

`maxpossiblescore` The maximum possible score.

`numquestions` The number of questions on the quiz.

`passingscore` The minimum score required to pass the quiz. Set by the quiz creator.

`sco-id` The ID of the quiz.

Contained tags

None (empty tag).

Returned by

[report-quiz-summary](#)

Example

```
<report-quiz-summary sco-id="217706" numquestions="5" maxpossiblescore="100"
  passingscore="0" />
```

report-quiz-takers

Description

Container tag; contains one or more `row` tags, each containing information about a user who has taken the specified course or quiz.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-course-takers](#), [report-quiz-takers](#)

Example

```
<report-quiz-takers>
  <row transcript-id="586821" sco-id="586760" principal-id="506385"
    status="not-attempted" score="0" percentage-score="0" time-taken=""
    certificate="">
    <name>Math Test</name>
    <login>benedict@example.com</login>
    <date-taken>2004-02-27T10:53:05.013-08:00</date-taken>
    <principal-name>Benedict, Fred</principal-name>
  </row>
  <row transcript-id="586837" sco-id="586760" principal-id="557527"
    status="user-failed" score="10" percentage-score="25"
    time-taken="51796" certificate="">
    <name>Math Test</name>
    <login>benedict2@example.com</login>
    <date-taken>2004-02-27T10:58:36.373-08:00</date-taken>
    <principal-name>Benedict, George</principal-name>
  </row>
</report-quiz-takers>
```

report-quotas

Description

Container tag; contains information about the account quotas.

Attributes

None.

Contained tags

quota

Returned by

[report-quotas](#)

Example

```
<report-quotas>
  <quota acl-id="7" quota-id="2" used="0" limit="unlimited" soft-
    limit="1000000000">
    <date-begin>1999-12-31T16:00:00.000-08:00</date-begin>
    <date-end>2999-12-31T16:00:00.000-08:00</date-end>
  </quota>
  <quota acl-id="7" quota-id="download-quota" used="0" limit="50000"
    soft-limit="50000">
    <date-begin>1999-12-31T16:00:00.000-08:00</date-begin>
    <date-end>2999-12-31T16:00:00.000-08:00</date-end>
  </quota>
</report-quotas>
```

report-sco-slides

Description

Container tag; contains one or more `row` tags, each indicating how many times and how recently a slide in a presentation has been viewed.

Attributes

None.

Contained tags

[row](#)

Returned by

[report-sco-slides](#)

Example

```
<report-sco-slides>
  <row slide="1" views="5">
    <last-viewed>2004-01-13T16:35:39.517-08:00</last-viewed>
  </row>
  <row slide="2" views="8">
    <last-viewed>2004-01-13T16:35:42.350-08:00</last-viewed>
  </row>
</report-sco-slides>
```

```

</row>
<row slide="3" views="7">
  <last-viewed>2004-01-13T16:35:44.933-08:00</last-viewed>
</row>
</report-sco-slides>

```

report-sco-views

Description

Container tag; contains one or more `row` tags, each indicating how many times and how recently the specified presentation has been viewed.

Attributes

None.

Contained tags

`row`

Returned by

`report-sco-views`

Example

```

<report-sco-views>
  <row sco-id="217706" type="presentation" is-folder="0" views="24">
    <name>Quiz test 1</name>
    <last-viewed-date>2004-01-13T17:11:47.923-08:00</last-viewed-date>
  </row>
</report-sco-views>

```

report-survey-question-response

Description

Container tag; contains information about the list of all users who answered the specified question and their answer.

Attributes

None.

Contained tags

`row`

Returned by

`report-survey-question-response`

Example

```

<report-survey-question-response>
  <row principal-id="8" question-number="1">
    <user-name>Enterprise Administrator</user-name>
    <question>how are you?</question>
  </row>
</report-survey-question-response>

```

```

        <response>fine</response>
    </row>
    <row principal-id="123" question-number="1">
        <user-name>test1 laxmi</user-name>
        <question>how are you?</question>
        <response>ok</response>
    </row>
</report-survey-question-response>

```

report-training-concurrent-users

Description

Indicates the largest number of users who have used the system at the same time.

Attributes

`account-id` The ID of the current account.

`limit` The maximum number of users who can use the system at the same time, as specified by your account license.

`max-num-sessions` The largest number of sessions that have ever been open simultaneously.

Contained tags

`name`

Returned by

`report-training-concurrent-users`

Example

```

<report-training-concurrent-users account-id="181224" max-num-sessions="2"
  limit="1000000000">
  <name>Professor Test Account</name>
</report-training-concurrent-users>

```

response

Description

A user's response to a survey question.

Attributes

None.

Contained tags

None (content only).

Returned by

`report-survey-question-response`

Example

```

<response>fine</response>

```

results

Description

Container tag; all results returned by any action are contained in a `results` tag.

Attributes

None.

Contained tags

Any.

Returned by

All actions.

Example

```
<results>
  <status code="ok" />
</results>
```

row

Description

Contains a single record of information in a report.

For examples of the possible attributes and contained tags, examine the sample results provided in the documentation for the action you're calling.

Attributes

The attributes for the `row` tag vary widely, depending on which action the tag is in response to.

Contained tags

The tags contained in the `row` tag vary widely depending on which action the tag is in response to and may include custom tags.

Returned by

report-course-takers, report-meeting-attendance, report-meeting-sessions, report-meeting-session-slots, report-quiz-answer-distribution, report-quiz-definition-answers, report-quiz-definition-questions, report-quiz-interactions, report-quiz-question-answer-distribution, report-quiz-question-distribution, report-quiz-question-response, report-quiz-takers, report-sco-slides, report-sco-views, report-bulk-content-slide-views, report-bulk-content-quiz, report-bulk-content-quiz-results, report-bulk-content-slide-views, report-bulk-course-quiz, report-bulk-course-results, report-bulk-meeting, report-bulk-meeting-attendance, report-meeting-sessions, report-meeting-session-slots, report-principal-list, report-survey-question-response

Example

```
<row sco-id="622852" num-participants="1" num-guests="0" num-polls="0">
  <name>My Meeting</name>
  <date-begin>2004-03-05T10:33:09.030-08:00</date-begin>
</row>
```

SCO

Description

Container tag; contains information about a SCO, which is a single Breeze document.

For examples of the possible attributes and contained tags, examine the sample results provided in the documentation for the action you're calling.

Attributes

The attributes for the `sco` tag vary widely, depending on what action the tag is in response to.

Contained tags

The tags contained in the `sco` tag vary widely, depending on what action the tag is in response to.

Returned by

[report-active-meeting-presenters](#), [report-active-meetings](#), [sco-contents](#), [sco-info](#), [sco-nav](#), [sco-search](#), [sco-shortcuts](#), [sco-update](#)

Example

```
<sco author-contact-id="185" folder-id="71" host-id="" learning-time="-1"
  sco-data-id="173" sco-id="235" source-sco-id="" status="active"
  tree-id="25" type="session">
  <date-begin>2003-06-30T19:33:38.317+00:00</date-begin>
  <date-created>2003-06-30T19:33:38.317+00:00</date-created>
  <date-end>2003-07-10T13:05:42.180+00:00</date-end>
  <date-modified>2003-06-30T19:33:38.317+00:00</date-modified>
  <name>another - 06/30/2003 12:33 PM</name>
  <url-path>/p62559548/</url-path>
</sco>
```

sco-author

Description

Container tag; contains information about the author of a SCO.

Attributes

None.

Contained tags

[email](#), [first-name](#), [last-name](#)

Returned by

[sco-info](#)

Example

```
<sco-author>
  <email>tenant@example.com</email>
  <first-name>Pat</first-name>
  <last-name>Tennant</last-name>
</sco-author>
```

sco-data

Description

Container tag; contains information about the underlying data file (such as the file that contains a presentation) used in a SCO.

Attributes

`byte-count` The file size of the SCO's data file, in bytes.

`height` The height of the SCO, in pixels.

`length` The duration of a presentation, in milliseconds.

`sco-data-id` The ID of the SCO's data file.

`slide-count` The number of slides in the SCO. A value of -1 indicates that the slide count doesn't apply; for example, if the SCO is a meeting rather than a presentation, the slide count is -1.

`version` The version number of the SCO.

`width` The width of the SCO, in pixels.

Contained tags

`date-created`, `date-modified`, `physical-path`

Returned by

`sco-info`

Example

```
<sco-data byte-count="3993" height="252" length="-1" sco-data-id="95654"
  slide-count="-1" version="1" width="380">
  <date-created>2004-01-26T20:47:05.653-07:00</date-created>
  <date-modified>2004-01-26T20:47:05.653-07:00</date-modified>
  <physical-path>222914/503227</physical-path>
</sco-data>
```

sco-name

Description

The name of a particular SCO, such as the name of a meeting.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-meeting-attendance](#)

Example

```
<sco-name>My Meeting</sco-name>
```

sco-nav

Description

Container tag; contains information about the folder hierarchy that contains the specified SCO.

Attributes

None.

Contained tags

[sco](#)

Returned by

[sco-nav](#)

Example

```
<sco-nav>
  <sco sco-id="181230" tree-id="181230" type="folder" depth="2">
    <name>User Meetings</name>
  </sco>
  <sco sco-id="181576" tree-id="181230" type="folder" depth="1">
    <name>tenant@example.com</name>
  </sco>
</sco-nav>
```

sco-search-info

Description

Container tag; contains one or more [sco](#) tags, each containing information about a SCO that matches the specified search text.

Attributes

None.

Contained tags

[sco](#)

Returned by

[sco-search](#)

Example

```
<sco-search-info>
  <sco sco-id="202014" folder-id="202008" type="presentation" slide-count="4"
    length="20000">
    <name>Quiz 1</name>
    <url-path>/p92628681</url-path>
    <date-created>2003-11-26T22:52:28.973-08:00</date-created>
    <date-modified>2003-11-26T22:52:37.867-08:00</date-modified>
    <first-name>Pat</first-name>
    <last-name>Tennant</last-name>
    <hit>1</hit>
    <hit>4</hit>
  </sco>
</sco-search-info>
```

SCOS

Description

Container tag; contains one or more `sco` tags, each containing information about a SCO inside the specified folder.

Attributes

None.

Contained tags

[sco](#)

Returned by

[sco-contents](#)

Example

```
<scos>
  <sco sco-id="505677" status="no-content" type="folder" is-folder="1">
    <name>PPT meeting tests</name>
    <date-modified>2004-01-29T06:22:49.157-08:00</date-modified>
  </sco>
  <sco sco-id="193496" status="active" type="meeting" is-folder="0">
    <name>My Name</name>
    <date-begin>2003-08-27T00:00:00.000-07:00</date-begin>
    <date-end>2003-08-27T01:00:00.000-07:00</date-end>
    <date-modified>2003-08-26T23:33:20.300-07:00</date-modified>
    <duration>1900-01-02T/+:00:00.000+00:00</duration>
  </sco>
</scos>
```

shortcuts

Description

Container tag; contains one or more `sco` tags, each containing information about a folder that contains content relevant to the logged-in user. For more information, see the [sco-shortcuts](#) action.

Attributes

None.

Contained tags

[sco](#)

Returned by

[sco-shortcuts](#)

Example

```
<shortcuts>
  <sco tree-id="181230" sco-id="181576" type="my-meetings">
    <domain-name>http://admin.breeze.example.com</domain-name>
  </sco>
  <sco tree-id="181227" sco-id="181412" type="my-content">
    <domain-name>http://admin.breeze.example.com</domain-name>
  </sco>
  ...
  [Other sco tags here.]
  ...
  <sco tree-id="181230" sco-id="181230" type="user-meetings">
    <domain-name>http://admin.breeze.example.com</domain-name>
  </sco>
</shortcuts>
```

source-sco

Description

Container tag; contains information about the SCO for the content of a course.

Attributes

None.

Contained tags

`source-sco`

Returned by

[sco-info](#)

Example

```
<source-sco>
  <source-sco author-contact-id="586761" folder-id="506334" host-id=""
    learning-time="" sco-data-id="586762" sco-id="586760" source-sco-id=""
    status="active" tree-id="181225" type="presentation">
    <date-created>2004-02-27T10:31:06.717-08:00</date-created>
    <date-modified>2004-02-27T10:52:58.217-08:00</date-modified>
    <description>Simple test presentation</description>
    <name>Math Test</name>
    <url-path>/p60802018/</url-path>
  </source-sco>
</source-sco>
```

status

Description

A status code returned by the Breeze server whenever an action is called. This section provides information about all the status codes and subcodes that Breeze can return.

Attributes

code The status of the action. The following values are possible for this attribute:

Value	Description
invalid	Indicates that a call is invalid in some way. If the code attribute is set to "invalid", the status tag contains an <code>invalid</code> tag that provides more detail about what's wrong.
no-access	Indicates that you don't have permission to perform the action. This can happen in several situations; the value of the subcode attribute provides more details. If you provide an invalid access key, there is no subcode value.
no-data	Indicates that there is no data available (in response to an action that would normally result in returning data). Usually indicates that there is no item with the ID you specified. To resolve the error, change the specified ID to that of an item that exists.
ok	Indicates that the action has completed successfully.
too-much-data	Indicates that the action should have returned a single result but is actually returning multiple results. For example, if there are multiple users with the same user name and password, and you call the <code>login</code> action using that user name and password as parameters, the system cannot determine which user to log you in as, so it returns a "too-much-data" error.

subcode Sometimes used to provide more detail of the status of the action. For example, **subcode** values are used to differentiate between different situations in which **code** is set to **no-access**. The following values are possible for this attribute:

Value	Description
account-expired	The customer account has expired.
denied	Based on the supplied credentials, you don't have access to perform the action.
no-login	You haven't logged in. To resolve the error, log in (using the <code>login</code> action) before you call the action. For more information, see the <code>login</code> action.
no-quota	The account limits have been reached or exceeded.
not-available	The required resource is unavailable.
not-secure	You must use SSL to call this action.
pending-activation	The account has not yet been activated.
pending-license	The account's license agreement has not been settled.
sco-expired	The course or tracking content has expired.
sco-not-started	The meeting or course has not started.

Contained tags

`invalid`

Returned by

All actions.

Example

```
<status code="ok" />
```

time-slot-begin

Description

The start date and time for a meeting session, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

`report-meeting-session-slots`

Example

```
<time-slot-begin>2004-02-14T06:50:00.000-08:00</time-slot-begin>
```

time-slot-end

Description

The end date and time for a meeting session, in ISO 8601 date format.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-meeting-session-slots](#)

Example

```
<time-slot-end>2004-02-14T07:00:00.000-08:00</time-slot-end>
```

time-taken

Description

The time a course was taken.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-course-results](#)

Example

```
<time-taken>00:34:23.011</time-taken>
```

tree

Description

Container tag; contains a tree that belongs to a specified account.

Attributes

`account-id` The ID of an account.

`tree-id` The ID of the tree.

`sco-id` The ID of the SCO.

`type` The type of principal. Possible values include "user", "group", and various kinds of **primary groups**: "admins", "authors", "live-admins", "course-admins", and "user-meetings". You can create and modify users and groups, but not primary groups.

Contained tags

[domain-name](#)

Returned by

[trees-info](#)

Example

```
<tree account-id="7" tree-id="14" sco-id="14" type="user-meetings">
  <domain-name>nagarajan.macromedia.com</domain-name>
</tree>
```

trees

Description

Container tag; contains the list of trees that belong to the specified account.

Attributes

None.

Contained tags

[tree](#)

Returned by

[trees-info](#)

Example

```
<trees>
  <tree account-id="7" tree-id="9" sco-id="9" type="content">
    <domain-name>nagarajan.macromedia.com</domain-name>
  </tree>
  <tree account-id="7" tree-id="10" sco-id="10" type="courses">
    <domain-name>nagarajan.macromedia.com</domain-name>
  </tree>
  <tree account-id="7" tree-id="11" sco-id="11" type="user-content">
    <domain-name>nagarajan.macromedia.com</domain-name>
  </tree>
  <tree account-id="7" tree-id="13" sco-id="13" type="meetings">
    <domain-name>nagarajan.macromedia.com</domain-name>
  </tree>
  <tree account-id="7" tree-id="14" sco-id="14" type="user-meetings">
    <domain-name>nagarajan.macromedia.com</domain-name>
  </tree>
</trees>
```

url

Description

A URL.

Attributes

None.

Contained tags

None (content only).

Returned by

`common-info`, `report-my-courses`

Example

```
<url>/api/xml?accesskey=yyyyyyyyyyyyyyyy&action=common-info</url>
```

url-path

Description

The path on the Breeze server to the specified SCO.

Attributes

None.

Contained tags

None (content only).

Returned by

`report-my-courses`, `report-my-meetings`, `sco-info`, `sco-search`

Example

```
<url-path>/r99718924/</url-path>
```

user (common-info)

Description

Container tag; contains information about a user.

Attributes

`user-id` The ID of the user.

Contained tags

`login`

Returned by

`common-info`

Example

```
<user user-id="503562">
  <login>gerweck@example.com</login>
</user>
```

user (user-accounts)

Description

Container tag; contains information about a user and an account.

Attributes

`user-id` The ID of the user.
`account-id` The ID of an account.

Contained tags

`date-expired`, `name`

Returned by

`user-accounts`

Example

```
<user user-id="211729" account-id="211655">
  <name>james@example.com</name>
  <date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
</user>
```

user-first-name

Description

A user's first name.

Attributes

None.

Contained tags

None.

Returned by

`report-bulk-content-slide-views`

Example

```
<user-first-name>Enterprise</user-first-name>
```

user-last-name

Description

A user's last name.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-bulk-content-slide-views](#)

Example

```
<user-last-name>Administrator</user-last-name>
```

user-login

Description

The login of a user.

Attributes

None.

Contained tags

None.

Returned by

[report-bulk-content-quiz-results](#)

Example

```
<user-login>lnagaraj@macromedia.com</user-login>
```

user-name

Description

A user name.

Attributes

None.

Contained tags

None (content only).

Returned by

[report-survey-question-response](#)

Example

```
<user-name>Enterprise Administrator</user-name>
```

users

Description

Container tag; contains one or more `user` tags, each containing information about a user and an account.

Attributes

None.

Contained tags

`user (user-accounts)`

Returned by

`user-accounts`

Example

```
<users>
  <user user-id="211729" account-id="211655">
    <name>james@example.com</name>
    <date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
  </user>
  <user user-id="212988" account-id="212928">
    <name>smith@example.com</name>
    <date-expired>2004-01-24T00:59:59.000-07:00</date-expired>
  </user>
</users>
```

value

Description

The value of a custom field.

Attributes

None.

Contained tags

None (content only).

Returned by

`principal-info`

Example

```
<value>408-1111-111</value>
```

version

Description

The Breeze version number.

Attributes

None.

Contained tags

None (content only).

Returned by

`common-info`

Example

```
<version>breeze_402_r116</version>
```

view-date-time

Description

The time and date that a slide was viewed.

Attributes

None.

Contained tags

None.

Returned by

`report-bulk-content-slide-views`

Example

```
<view-date-time>2004-06-30T11:46:12.500-07:00</view-date-time>
```


INDEX

A

- access keys
 - acquiring 42
 - Breeze Enterprise customers 18
 - hosted customers 18
 - overview 17
 - special 41, 119, 120, 121
- access-key tag 119, 120
- access-keys tag 121
- accesskey-exec action 41
- accesskey-info action 42
- account tag 121
- accounts
 - limits 74
 - listing 110
- acl-id parameters 17
- acl-preference-update action 43
- ACLs 18
- action tag 122
- action-list action 46
- actions
 - accesskey-exec 41
 - accesskey-info 42
 - acl-preference-update 43
 - action-list 46
 - alphabetical listing 41
 - calling 15
 - common-info 33, 47
 - completed successfully 29
 - custom-field-update 47
 - custom-fields 48
 - custom-fields-delete 49
 - defined 9
 - group-membership-update 50
 - information updates 9
 - login 29, 50
 - logout 52
 - names 122
 - permissions-info 52
 - permissions-reset 54
 - permissions-update 54
 - principal-info 32, 55
 - principal-list 30, 56
 - principal-update 31, 57
 - principals-delete 31, 59
 - reference entries 41
 - report-account-meeting-attendance 59
 - report-active-meeting-presenters 61
 - report-active-meetings 61
 - report-bandwidth 62
 - report-bulk-content-quiz-results 64
 - report-bulk-content-slide-views 65
 - report-bulk-course-quiz 66
 - report-bulk-course-results 67
 - report-bulk-meeting 68
 - report-bulk-meeting-attendance 69
 - report-course-status 70
 - report-course-takers 35, 71
 - report-disk-usage 72
 - report-meeting-attendance 73
 - report-meeting-concurrent-users 74
 - report-meeting-session 75
 - report-meeting-session-slots 76
 - report-meeting-sessions 75
 - report-meeting-summary 77
 - report-my-courses 33, 78
 - report-my-meetings 33, 79
 - report-principal-list 80
 - report-quiz-definition-answers 82
 - report-quiz-definition-questions 83
 - report-quiz-interactions 84
 - report-quiz-question-answer-distribution 86
 - report-quiz-question-distribution 87
 - report-quiz-question-response 88

- report-quiz-question-totals 89
- report-quiz-summary 90
- report-quiz-takers 90
- report-quotas 92
- report-sco-slides 93
- report-sco-views 94
- report-survey-question-response 94
- report-training-concurrent-users 95
- report-quiz-answer-distribution 81
- sco-build 96
- sco-contents 34, 97
- sco-delete 98
- sco-info 34, 99
- sco-move 101
- sco-nav 101
- sco-search 102
- sco-shortcuts 33, 103
- sco-update 104
- sco-upload 108
- trees-info 109
- user-accounts 110
- user-update-pwd 111
- See also* web services
- actions tag 122
- active meetings
 - listing 61
 - presenters 61
- adding
 - principals to a group 50
 - users to Breeze 31
- administrative users 29
- admins primary group 16
- answer-text tag 123
- answers to quiz questions
 - answered correctly 87
 - distribution 86
 - interactions 84
 - text of 123
- architecture overview 13
- archived meetings 17
- attendance
 - at meetings 73
 - maximum 74
- attributes, searching 30
- authors primary group 16

B

- bandwidth, reporting 62
- basics of web services 10
- body tag 123

- bold text convention 11
- Breeze
 - displaying relevant content 33
 - documentation 9
 - integrating with a portal 32
 - interacting with the server 14
 - users, adding 31, 41
 - users, deleting 31
 - users, updating 32
 - version number 185
- BREEZESession cookies
 - code sample 29
 - cookie name conflicts 20
 - cookie tag 125
 - invalidating 52
 - overview 17
 - security 17
- building SCOs 96, 108

C

- calling actions 14, 15, 27
- cfhttp tag (in CFML) 30
- CFML
 - code samples 15
 - using with XML 10
- changing
 - group membership 50
 - permissions 54
 - principals 57
- checking for errors 29
- code font convention 11
- code samples
 - overview 27
 - parsing returned XML code 29
 - use of CFML 15
- ColdFusion Markup Language. *See* CFML
- comma-separated values (CSV) 35
- common tag 124
- common tasks 27
- common-info action 33, 47
- configuring access keys 18
- contact tag 125
- content
 - folder 97
 - launching 20
 - relevant, displaying 33
 - searching 35
 - URLs, determining 34
- conventions, typographical 11
- cookie tag 125

- cookies. *See* BREEZESESSION cookies
- course tag 126
- course-admins primary group 16
- courses
 - about 16
 - enrolled students 71
 - launching 20
 - listing 33, 35, 78
 - See also* SCOs
- creating
 - administrative users 29
 - meetings 35
 - principals 57
 - query strings 15
 - SCOs 104
 - users 31
- credentials, security 17, 50
- CSV format 35
- custom-field-update action 47
- custom-fields action 48
- custom-fields tag 126
- custom-fields-delete action 49

D

- data flow 14
- date formatting 22
- date tag 127
- date-begin tag 127
- date-closed tag 128
- date-created tag 128
- date-end tag 128
- date-expired tag 129
- date-last-taken tag 129
- date-modified tag 129
- date-taken tag 130
- date-time-attempted tag 130
- dates and times 10
- deleting
 - principals 59
 - SCOs 98
 - users 31
- denied permission 19
- description tag 131
- descriptions of principals 55
- directory services, synchronizing 28
- disk space used 72
- displaying returned information in a table 34
- documentation and updates 9

- domain names, determining 34
- domain-name tag 131
- duration tag 131

E

- e-mail addresses as login names 136
- email tag 132
- enrolled students in a course 71
- error-checking 29
- expired tag 132

F

- field tag 132
- filters 21
 - examples 22, 23
 - reference 24
 - sort 23
 - testing in browser 24
- finding
 - courses 33
 - meetings 33
 - text in SCOs 102
- first-name tag 133
- Flash files 16
- folder-id parameters 17
- folders

- about 16
 - common 103
 - contents 97
 - hierarchies 101
 - listing contents 34
 - my-content 33
 - permissions 19
 - relevant to current user 103
- See also* SCOs

G

- generating reports 35
- GET and POST requests (HTTP) 15, 30
- group-id parameters 17
- group-membership-update action 50
- groups
 - about 16
 - membership, updating 50
 - permissions 19
 - primary 16
 - See also* principals, users
- guides 9

H

- help systems 9
- hierarchies of folders 101
- hit tag 134
- host tag 134
- hosted accounts, access keys for 18
- HTTP
 - GET and POST requests 15
 - parameters 30

I

- identity verification 50
- IDs
 - about 16
 - account 121
 - permissions 18
 - quiz questions 88
 - user's, determining 33
- image files 16
- information
 - about principals 55
 - about SCOs 99
 - about servers, obtaining 47
 - displaying 34
- inserting a user into Breeze 31
- instructional media 9
- integrating a portal with Breeze 32
- interacting with the Breeze server 14
- interactions with quizzes 84
- interfaces, exposed 13
- introduction to web services 10
- invalid tag 134
- invitees, number of 77
- ISO 8601 date format 10
- italics convention 11

J

- JSESSIONID cookies
 - login action 50

K

- keys, access 17

L

- last-name tag 135
- last-viewed tag 136
- last-viewed-date tag 136
- late-breaking news 9
- launching content 20

- LDAP 28

- licensed customers, access keys for 18
- Lightweight Directory Access Protocol (LDAP) 28
- limits on accounts 74
- links, creating 34
- listing
 - accounts 110
 - courses 33, 78
 - folder contents 34, 97
 - meetings 33, 61
 - presenters 61
 - principals 56
 - quiz questions 83
 - students 71
- live-admins primary group 16
- logging in 17, 29, 32, 50
- logging out 52
- login action 17, 50
- login names, determining 55
- login tag 136
- logos 17
- logout action 52

M

- manage permission 19
- manuals 9
- maximum meeting attendance 74
- meeting tag 137
- meetings
 - about 16
 - active 61
 - archived 17
 - creating 35
 - empty 75
 - invitees, number of 77
 - launching 20
 - listing 33, 61
 - permissions 19
 - presenters 61
 - scheduled 79
 - schedules for meetings 75
 - sessions 75
 - See also* SCOs
- meetings, attendance
 - attendees in ten-minute period 76
 - list of users 73
 - maximum 74
 - number of attendees 75
 - summary 77
- moving SCOs 101

- multiple parameters, specifying
 - overview 15
 - permissions update 54
 - principals, deleting 59
 - SCOs, deleting 98
- my-content folder 33
- my-courses tag 137
- my-meetings tag 138

N

- name tag 139
- names of actions 122
- new information 9

O

- online resources 10
- optional parameters 15

P

- parameters
 - about 15
 - ID 17
 - multiple, specifying 15, 54, 59, 98
 - optional 15
 - session 20
 - specifying 15
- parent folders 101
- parent-acl-id parameters 17
- parsing returned XML code 29
- participant-name tag 139
- passwords
 - changing 111
 - checking 33
- paths to content 34
- permissions
 - about 18
 - assigning 105
 - public access 20
 - See also* SCOs
- permissions tag 139, 140
- permissions-info action 52
- permissions-reset action 54
- permissions-update action 54
- physical-path tag 141
- Pods 17
- portals, integrating Breeze with 32
- POST requests (HTTP) 15, 30
- prerequisites 9

- presentations
 - about 16
 - number of slides 173
 - number of times viewed 93, 94
 - searching 35
 - See also* SCOs
- presenters
 - of meetings 61
 - permissions 19
- primary groups
 - defined 16
 - permissions 18
- principal tag 144
- principal-id parameters 17
- principal-info action 32, 55
- principal-list action 30, 56
- principal-list tag 146
- principal-name tag 146
- principal-update action 31, 57
- principals
 - about 16
 - adding to a group 50
 - creating 57
 - information, obtaining 55
 - listing 56
 - permissions 18, 52, 54
 - public-access 20
 - See also* users, groups
- principals-delete action 31, 59
- public access permissions 20
- publish permission 19

Q

- query strings, building 15
- question tag 147
- question-text tag 147
- questions
 - IDs, determining 88
 - listing 83
- quiz-definition-answers tag 148
- quiz-definition-questions tag 148
- quiz-name tag 149
- quizzes
 - answers, distribution of 86
 - answers, text of 123
 - dates taken 130
 - interactions 84
 - number of answering users 89
 - questions 83
 - questions answered correctly 87

- scoring 84
- summaries of results 90
- users who took a quiz 90
- See also* SCOs
- quota tag 149

R

- rebuilding SCOs 96
- recent updates to documentation 9
- registering a user 41
- removing
 - principals from a group 50
 - principals from Breeze 59
 - users 31
- report-account-meeting-attendance action 59
- report-account-meeting-attendance tag 150
- report-active-meeting-presenters action 61
- report-active-meeting-presenters tag 151
- report-active-meetings action 61
- report-active-meetings tag 151
- report-bandwidth action 62
- report-bandwidth tag 152
- report-bulk-content-quiz tag 152
- report-bulk-content-quiz-results action 64
- report-bulk-content-quiz-results tag 153
- report-bulk-content-slide-views action 65
- report-bulk-course-quiz action 66
- report-bulk-course-quiz tag 154
- report-bulk-course-quiz-results tag 154
- report-bulk-course-results action 67
- report-bulk-meeting action 68
- report-bulk-meeting tag 155
- report-bulk-meeting-attendance action 69
- report-bulk-meeting-attendance tag 156
- report-bulk-slide-views tag 156
- report-course-status action 70
- report-course-status tag 157
- report-course-takers action 35, 71
- report-disk-usage action 72
- report-disk-usage tag 158
- report-meeting-attendance action 73
- report-meeting-attendance tag 158
- report-meeting-concurrent-users action 74
- report-meeting-concurrent-users tag 159
- report-meeting-session action 75
- report-meeting-session tag 159
- report-meeting-session-slots action 76
- report-meeting-session-slots tag 160
- report-meeting-sessions action 75
- report-meeting-sessions tag 160

- report-meeting-summary action 77
- report-meeting-summary tag 161
- report-my-courses action 33, 78
- report-my-meetings action 33, 79
- report-principal-list action 80
- report-principal-list tag 161
- report-quiz-answer-distribution tag 162
- report-quiz-definition-answers action 82
- report-quiz-definition-questions action 83
- report-quiz-interactions action 84
- report-quiz-interactions tag 163
- report-quiz-question-answer-distribution action 86
- report-quiz-question-answer-distribution tag 163
- report-quiz-question-distribution action 87
- report-quiz-question-distribution tag 164
- report-quiz-question-response action 88
- report-quiz-question-response tag 165
- report-quiz-question-totals action 89
- report-quiz-question-totals tag 165
- report-quiz-results tag 166
- report-quiz-summary action 90
- report-quiz-summary tag 166
- report-quiz-takers action 90
- report-quiz-takers tag 167
- report-quotas action 92
- report-quotas tag 168
- report-sco-slides action 93
- report-sco-slides tag 168
- report-sco-views action 94
- report-sco-views tag 169
- report-survey-question-response action 94
- report-survey-question-response tag 169
- report-training-concurrent-users action 95
- report-training-concurrent-users tag 170
- reports
 - CSV format 35
 - generating 35
- report-quiz-answer-distribution action 81
- requesting a list of Breeze users 30
- required parameters 15
- resetting permissions 54
- resources
 - bandwidth, reporting 62
 - disk space, reporting 72
 - documentation, additional 10
- response tag 170
- results
 - of actions 113
 - of quizzes 90

- results tag 113, 171
- returned XML elements 37, 113
- row tag 35, 171

S

- sample code 27
- scenarios, common 27
- scheduled meetings 79
- sco tag 172
- sco-author tag 172
- sco-build action 96
- sco-contents action 34, 97
- sco-data tag 173
- sco-delete action 98
- sco-id parameters 17
- sco-info action 34, 99
- sco-move action 101
- sco-name tag 173
- sco-nav action 101
- sco-nav tag 174
- sco-search action 102
- sco-search-info tag 174
- sco-shortcuts action 33, 103
- sco-update action 104
- sco-upload action 108
- scores on quizzes 84
- SCOs
 - about 16
 - building 96, 108
 - commonly used 103
 - creating 104
 - deleting 98
 - expiration, defined 132
 - folder hierarchies 101
 - information about, acquiring 99
 - listing in a folder 97
 - moving 101
 - parent folders 101
 - parent permissions 19
 - permissions, information about 52
 - permissions, overview 18
 - permissions, parent 19
 - permissions, resetting 54
 - permissions, updating 54
 - searching 35, 102
 - updating 104
 - uploading 108
 - URLs, determining 103
 - See also* courses, folders, meetings, presentations, quizzes

- scos tag 175
- searching
 - for content 35
 - SCOs 102
- security overview 17
- self-registering 41, 119, 120, 121
- servers, information about 47
- session query parameter 20
- sessions of meetings 75, 76
- setting permissions 54
- shortcuts 33, 103
- shortcuts tag 176
- slides
 - number of slides 173
 - number of times viewed 93
- slots for meeting sessions 76
- SOAP (Simple Object Access Protocol) 13
- sort filters 23
- source-sco tag 176
- special access keys 41
- status codes
 - invalid 134
 - overview 177
- status tag 29, 113, 177
- students, listing 71
- subfolders, listing 97
- summaries of quiz results 90
- synchronizing with directory services 28

T

- tables, displaying 34
- tags
 - access-key 119, 120
 - access-keys 121
 - account 121
 - action 122
 - actions 122
 - answer-text 123
 - body 123
 - common 124
 - contact 125
 - cookie 125
 - course 126
 - custom-fields 126
 - date 127
 - date-begin 127
 - date-closed 128
 - date-created 128
 - date-end 128
 - date-expired 129

- date-last-taken 129
- date-modified 129
- date-taken 130
- date-time-attempted 130
- description 131
- domain-name 131
- duration 131
- email 132
- expired 132
- field 132
- first-name 133
- hit 134
- host 134
- invalid 134
- last-name 135
- last-viewed 136
- last-viewed-date 136
- login 136
- meeting 137
- my-courses 137
- my-meetings 138
- name 139
- overview 113
- participant-name 139
- permissions 139, 140
- physical-path 141
- principal 144
- principal-list 146
- principal-name 146
- question 147
- question-text 147
- quiz-definition-answers 148
- quiz-definition-questions 148
- quiz-name 149
- quota 149
- report-account-meeting-attendance 150
- report-active-meeting-presenters 151
- report-active-meetings 151
- report-bandwidth 152
- report-bulk-content-quiz 152
- report-bulk-content-quiz-results 153
- report-bulk-course-quiz 154
- report-bulk-course-quiz-results 154
- report-bulk-meeting 155
- report-bulk-meeting attendance 156
- report-bulk-slide-views 156
- report-course-status 157
- report-disk-usage 158
- report-meeting-attendance 158
- report-meeting-concurrent-users 159
- report-meeting-session 159
- report-meeting-session-slots 160
- report-meeting-sessions 160
- report-meeting-summary 161
- report-principal-list 161
- report-quiz-answer-distribution 162
- report-quiz-interactions 163
- report-quiz-question-answer-distribution 163
- report-quiz-question-distribution 164
- report-quiz-question-response 165
- report-quiz-question-totals 165
- report-quiz-results 166
- report-quiz-summary 166
- report-quiz-takers 167
- report-quotas 168
- report-sco-slides 168
- report-sco-views 169
- report-survey-question-response 169
- report-training-concurrent-users 170
- response 170
- results 113, 171
- row 35, 171
- sco 172
- sco-author 172
- sco-data 173
- sco-name 173
- sco-nav 174
- sco-search-info 174
- scos 175
- shortcuts 176
- source-sco 176
- status 113, 177
- time-slot-begin 178
- time-slot-end 179
- tree 179, 180
- url 181
- url-path 181
- user 181, 182
- user-first-name 182
- user-last-name 183
- user-login 183
- user-name 183
- users 184
- value 184
- version 185
- view-date-time 185
- tasks, common 27
- text, searching for 102
- time-slot-begin tag 178
- time-slot-end tag 179

- times and dates 10
- tree tag 179
- trees 16
- trees tag 180
- trees-info action 109
- troubleshooting 177
- typographical conventions 11

U

- updates to the documentation 9
- updating

- group membership 50
 - passwords 111
 - permissions 54
 - principals 57
 - SCOs 104
 - user information 32

- uploading SCOs 96, 108

- url tag 181

- url-path tag 181

- URLs, determining 34, 103

- use cases, common 27

- user tag 181, 182

- user-accounts action 110

- user-first-name tag 182

- user-id parameters 17

- user-last-name tag 183

- user-login tag 183

- user-name tag 183

- user-update-pwd action 111

- users

- about 16
 - accounts, listing 110
 - adding 31
 - administrative 29
 - courses, enrolled in 33, 78
 - courses, taking 71
 - deleting 31
 - folders, relevant 103
 - groups, changing membership 50
 - IDs, determining 33
 - information, obtaining 47, 55
 - listing 30
 - logging in 32, 50
 - logging out 52
 - maximum concurrent 95
 - meeting attendance, maximum 74
 - meetings, attending 73
 - meetings, scheduled 33, 79
 - passwords, changing 111

- permissions 52

- quiz takers 90

- registering with Breeze 41

- updating information for 32

- See also* principals, groups

- users tag 184

V

- value tag 184

- verifying identity 50

- version tag 185

- view permission 19

- view-date-time tag 185

- view-hidden permission 20

- view-only permission 20

- viewing

- presentations 94

- slides 93

W

- web services

- architecture overview 13

- primer 10

- updates 9

- See also* actions

- workflow 14

X

- XML results

- overview 113

- parsing 29

- XML web services. *See* actions, web services

- XML, using with CFML 10

- XML_API_KEY configuration 18

- XSLT (XSL Transformations) 10, 35

